

AD-A061 307

INTERNATIONAL COMPUTING CO BETHESDA MD
PRELIMINARY DESIGN STUDY FOR VTS PROCESSING/DISPLAY SUBSYSTEM.(U)
JUN 78 C C HENSON, R A CLEAVER, S H KAISLER DOT-CG-81-78-1833
USCG-D-66-78 NL

UNCLASSIFIED

1 OF 44
AD
A061307



REPORT NO. CG-D-66-78

LEVEL II

PRELIMINARY DESIGN STUDY
FOR
VTS PROCESSING/DISPLAY SUBSYSTEM

Carle C. Henson
Ronald A. Cleaver
Stephen H. Kaisler

INTERNATIONAL COMPUTING COMPANY
4330 EAST-WEST HIGHWAY
BETHESDA, MARYLAND 20014



JUNE 1978

FINAL REPORT

Document is available to the U.S. public through
the National Technical Information Service
Springfield, Virginia 22161

PREPARED FOR
U.S. DEPARTMENT OF TRANSPORTATION
UNITED STATES COAST GUARD
OFFICE OF RESEARCH AND DEVELOPMENT
WASHINGTON, D.C. 20590

78 11 01 07

AD A061307

DDC FILE COPY

D
R
NOV
R
L

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

The contents of this report reflect the views of the Coast Guard Research and Development Center, which is responsible for the facts and accuracy of data presented. This report does not constitute a standard, specification, or regulation.

D. L. Birkimer

DONALD L. BIRKIMER, Ph.D., P.E.
Technical Director
U.S. Coast Guard Research and Development Center
Avery Point, Groton, Connecticut 06340

18 USCG, CGR/DC

Technical Report Documentation Page

1. Report No. CG-D-66-78, 15/78		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Preliminary Design Study for VTS Processing/Display Subsystem		5. Report Date June 1978		6. Performing Organization Code	
7. Author(s) Carle C./Henson, Ronald A./Cleaver, Stephen H./Kaisler		8. Performing Organization Report No. 12-313p.		9. Performing Organization Name and Address International Company Company* 4330 East-West Highway Bethesda, Maryland 20014	
10. Sponsoring Agency Name and Address U. S. Department of Transportation United States Coast Guard Office of Research and Development Washington, DC 20590		11. Work Unit No. (TRAIS) 4502.2.1.4		12. Contract or Grant No. DOT-CG-81-78-1833	
13. Type of Report and Period Covered Final Report. January to June 1978		14. Sponsoring Agency Code		15. Supplementary Notes The contract under which this report was submitted was under the technical supervision of the Coast Guard Research and Development Center, Groton, Connecticut 06340. R&D Center report number CGR/DC 15/78 has been assigned.	
16. Abstract Distributed processing concepts are applied to the design of a highly reliable, modular multiple computer system. A preliminary model of processing and data base requirements is presented for a range of potential Vessel Traffic Services (VTS) configurations. A survey of system architectures and the state-of-the-art in distributed processing is included. Classes of processors, interconnections and logical structures are considered. Display technology is surveyed. Based on the system models and technology surveys, four system architectures are selected which include: large minis and small mini/16bit micros; interconnection by shared bus and by a network of direct serial lines; and alternative logical structures. A preliminary design is presented for each candidate architecture. A set of evaluation criteria and an evaluation methodology is presented. The four candidate architectures are evaluated on the basis of simplicity, feasibility, modularity/flexibility, maintainability, expandability, reliability, cost and performance. Two architectures are recommended for further consideration. Both use relatively large minis interconnected by a shared bus. The lesser rated of the two maintains functional assignments of processors throughout the VTS configurations studied. The highest rated architecture allows functional assignments to be rearranged from one configuration to another, thus reducing the hardware required.					
17. Key Words Vessel Traffic Services (VTS) Synthetic Displays Distributed Processing System Architecture Multicomputer Systems			18. Distribution Statement Document is available to the U. S. Public through the National Technical Information Service, Springfield, Virginia 22161		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 310	
				22. Price	

Form DOT F 1700.7 (8-72) Reproduction of form and completed page is authorized

SK 393 469

LB

METRIC CONVERSION FACTORS

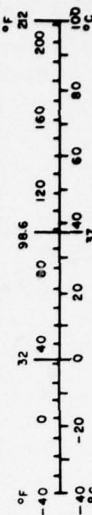
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
AREA				
in ²	square inches	6.5	square centimeters	cm ²
ft ²	square feet	0.09	square meters	m ²
yd ²	square yards	0.8	square meters	m ²
mi ²	square miles	2.6	square kilometers	km ²
	acres	0.4	hectares	ha
MASS (weight)				
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
VOLUME				
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft ³	cubic feet	0.03	cubic meters	m ³
yd ³	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 m = 2.54 inches. For other exact conversions and more detail, see NBS Mon. Publ. 286, Units of Weights and Measures, Price \$2.25, SD Catalog No. C13.10.286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
km	kilometers	1.1	yards	yd
		0.6	miles	mi
AREA				
cm ²	square centimeters	0.16	square inches	in ²
m ²	square meters	1.2	square yards	yd ²
km ²	square kilometers	0.4	square miles	mi ²
ha	hectares (10,000 m ²)	2.5	acres	
MASS (weight)				
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
VOLUME				
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
m ³	liters	0.26	gallons	gal
m ³	cubic meters	35	cubic feet	ft ³
m ³	cubic meters	1.3	cubic yards	yd ³
TEMPERATURE (exact)				
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



CONTENTS

ACCESSION NO.	<input checked="checked" type="checkbox"/> White Section <input type="checkbox"/> Buff Section	UNANNOUNCED JUSTIFICATION	BY DISTRIBUTION/AVAILABILITY CODES	DATE AVAIL. DATE/REVISION	A
---------------	---	------------------------------	---------------------------------------	------------------------------	---

<u>Section</u>		<u>Page</u>
1	INTRODUCTION	1-1
1.1	Purpose	1-1
1.2	Scope	1-2
1.3	Interpretation of Preliminary Results	1-5
1.4	Organization	1-6
2	OVERVIEW OF VTS PROCESSING/DISPLAY SUBSYSTEM REQUIREMENTS	2-1
2.1	Introduction	2-1
2.2	Purpose	2-1
2.3	VTC Functions and Organization	2-2
2.4	VTS System Modes, Classes and Sensor Levels	2-4
2.5	VTS System Design Goals	2-7
2.6	VTS System Data Base	2-9
2.7	Processing Capabilities	2-11
3	PROCESSING REQUIREMENTS	3-1
3.1	Introduction	3-1
3.2	Approach	3-3
3.3	Processing Categories	3-6
3.4	First Approximation	3-11
3.5	Second Approximation	3-40
3.6	Summary	3-57
4	DATA BASE REQUIREMENTS	4-1
4.1	Introduction	4-1
4.2	Access Methods	4-2
4.3	File Structures and Sizes	4-5
4.4	Disc Access Frequencies	4-21

CONTENTS

<u>Section</u>		<u>Page</u>
5	DISPLAY SYSTEMS ANALYSIS	5-1
5.1	VTS Display Station Requirements	5-2
5.2	Survey of Display Technology	5-21
5.3	Design Assumptions	5-43
6	FEASIBLE ARCHITECTURES	6-1
6.1	Perspective	6-1
6.2	What is a System Architecture?	6-3
6.3	Processors	6-4
6.4	Interconnection	6-7
6.5	Logical Structure	6-21
7	SELECTION OF ARCHITECTURES	7-1
7.1	Logical and Physical Characteristics	7-1
7.2	Development of Candidate Architectures	7-5
8	ARCHITECTURE EVALUATION AND SELECTION CRITERIA	8-1
8.1	Simplicity	8-3
8.2	Feasibility	8-6
8.3	Modularity/Flexibility	8-9
8.4	Maintainability	8-12
8.5	Expandability	8-16
8.6	Reliability	8-18
8.7	Cost	8-22
8.8	Performance	8-26
9	CANDIDATE ARCHITECTURE I	9-1
9.1	Physical Structure	9-2
9.2	Logical Structure	9-3
9.3	Class C, Level 4 System	9-4
9.4	Class B, Level 4 System	9-23
9.5	Class A, Level 1 System	9-34
9.6	Relative Hardware Costs	9-39

CONTENTS

		<u>Page</u>
10	CANDIDATE ARCHITECTURE II	10-1
10.1	Physical Structure	10-1
10.2	Logical Structure	10-2
10.3	Class C, Level 4 System	10-3
10.4	Class B, Level 4 System	10-13
10.5	Class A, Level 1 System	10-21
10.6	Relative Hardware Costs	10-24
11	CANDIDATE ARCHITECTURE III	11-1
11.1	Class C, Level 4 System	11-1
11.2	Class B, Level 4 System	11-4
11.3	Class A, Level 1 System	11-6
12	CANDIDATE ARCHITECTURE IV	12-1
12.1	Physical Structure	12-1
12.2	Logical Structure	12-2
12.3	Class C, Level 4 System	12-3
12.4	Class B, Level 4 System	12-13
12.5	Class A, Level 1 System	12-15
12.6	Relative Hardware Costs	12-15
13	COMPARATIVE EVALUATION OF ARCHITECTURES	13-1
13.1	Evaluation Methodology	13-2
13.2	Discussion of Weighting Factors	13-7
APPENDIX	HARDWARE COST FACTORS	A-1

1.1 PURPOSE

This report was prepared by International Computing Company (ICC) under Contract No. DOT-CG-81-78-1833, for the United States Coast Guard (USCG). This report presents the results of a preliminary hardware and software architecture design study for the Vessel Traffic Services (VTS) processing/display subsystem.

Significant design constraints established by the USCG for consideration in this study are:

- . Use of a distributed processing system concept
- . Use of multiple computer architecture
- . Modularity of system design
- . High reliability (99.9% availability)
- . Use of off-the-shelf hardware

No restrictions, however, were placed on the size (i.e., speed) of the processors, the interconnection mechanism for the multiple processors, or the level of redundancy required to achieve the desired availability. For each of the above parameters, we will examine several alternatives in an effort to define a practical system, both in terms of system performance and cost.

1.2 SCOPE

This report concludes Phase I of the three phase design effort for this VTS subsystem. The three phases are:

- . Phase I - preliminary system design and identification of two feasible system architectures.
- . Phase II - detailed systems design based on the architecture chosen by the USCG from the two candidates identified in Phase I.
- . Phase III - development of detailed software requirements and software design specifications.

Phase I required the completion of eleven technical tasks:

- . Task 1 - VTS Familiarization and Planning.
This task allowed ICC to become familiar with the current VTS concept, functional specifications and system development requirements. Three harbors were visited. Two of the harbors currently have automated VTS systems (New Orleans and Houston), and one of the harbors will soon have an automated system (New York).
- . Task 2 - Performance Analysis.
This analysis was done to develop first approximations to the traffic, resource and processing requirements of the VTS Processing/Display Subsystem.

. Task 3 - Identify Feasible Architectures.

This task involved a survey of the state-of-the-art in distributed, multiple computer systems to identify candidate architectures for further analysis.

. Task 4 - Display Systems Analysis.

The objective of this task was to identify and describe suitable display systems for integration into the system architecture, and describe major hardware interconnections. In connection with this task a survey of current display technology and man/machine communication considerations was undertaken.

. Task 5 - Computing Requirements Analysis.

This task refined and expanded the approximations developed in Task 2. Overall estimates were developed for processing load, memory requirements, mass storage capacity and data base access rate.

. Task 6 - Selection of Feasible Architectures.

This task selected the four most promising architectures for further analysis.

. Task 7 - Architecture Analysis.

The objective of this task was to analyze in greater detail the four feasible architectures selected in order eventually to select the two most feasible. The results of Task 5 were re-evaluated and refined.

. Task 8 - Architecture Selection.

The objective of this task was to compare the results of each architecture analysis, and select two architectures which best satisfy the requirements of the VTS Processing/Display Subsystem.

. Task 9 - Selection Criteria.

This task enabled the final selection of two system architectures based on specific and detailed evaluation criteria and a selection methodology.

. Task 10 - Preliminary Design Report.

This task required production of a draft report for USCG review prior to delivery of the final report.

. Task 11 - Phase I Final Report.

This task involved the incorporation of USCG corrections or revisions and preparation of the final version of the Phase I Report.

1.3 INTERPRETATION OF PRELIMINARY RESULTS

This report presents quantitative results for the following system descriptors:

- . Processing load
- . Data base capacity
- . Data base access rates
- . Interprocessor communications requirements

These results are based on many assumptions made in an effort to compare system architectures. These results must not be construed as final or definitive, but rather should be considered preliminary results subject to revision as the need arises. Detailed systems requirements will be developed in Phases II and III, after judgments are made regarding the best system architecture for the VTS Processing/Display Subsystem, and regarding the reasonability of assumptions.

1.4 ORGANIZATION

This report is organized into thirteen sections:

- . Section 1 - Introduction
- . Section 2 - Overview of VTS Processing/Display
Subsystem Requirements
- . Section 3 - Processing Requirements
- . Section 4 - Data Base Requirements
- . Section 5 - Display Systems Analysis
- . Section 6 - Feasible Architectures
- . Section 7 - Criteria for Architecture Evaluation
- . Section 8 - Selected Architectures
- . Sections 9 - 12 - Architecture Analysis for Four
Selected Architectures
- . Section 13 - Architecture Comparison and Selection
of Two Architectures

2 OVERVIEW OF VTS PROCESSING/DISPLAY SUBSYSTEM REQUIREMENTS

2.1 INTRODUCTION

The USCG has developed a functional description for the Processing/Display subsystem of the modular, computerized VTS system. This documentation of system requirements is the first step in a classical system design effort. Documentation of system requirements provides a basis for common understanding of system methods and functions, and establishes a framework from which detailed design specifications may be developed, from which an integrated system may be built, and against which system adherence to requirements may be measured.

2.2 PURPOSE

This section provides an overview and summary of VTS Processing/Display Subsystem requirements. For additional details refer to Appendix 8 of Request for Proposal (RFP) No. 81-77-1833 issued by the United States Coast Guard Academy, New London, Connecticut, 06320.

The fundamental purpose of the VTS system is to reduce the number of accidents which occur in harbors by providing information with which vessel traffic may be managed, and by providing usable and reliable information on potential vessel hazards in a timely fashion.

2.3 VTC FUNCTIONS AND ORGANIZATION

According to Appendix 8, referenced above, the Vessel Traffic Center (VTC) has some or all of the following five major functions.

- . Detect and report potentially hazardous situations, such as imminent collisions, groundings, and excessive traffic congestion
- . Schedule and/or route traffic
- . Provide safety related information to the maritime community, such as other vessel traffic routes, and buoys off station
- . Maintain a traffic history for the harbor to assist with traffic research and enable measurement of VTS effectiveness
- . Maintain a log of all VTC activities for measurement of VTS effectiveness, and provide a detailed accounting of the activities of the VTC and participating vessels on occasions when accidents occur

The VTC is basically a data communications and processing network. It consists of:

- . A voice communications network enabling communications between watchstanders (i.e., system operators) and vessel pilots;
- . A network of sensors used to obtain information on vessel position and course, as well as waterway environmental parameters such as weather, tide and current data;

- . Data communications links between the sensors and the VTC;
- . An information storage, processing, retrieval, and display system;
- . A set of vessel traffic monitoring and management procedures for real-time analysis and management of traffic flow.

2.4 VTS SYSTEM MODES, CLASSES AND SENSOR LEVELS

The various harbors in which VTS systems could be installed offer a range of requirements which the system design must be capable of adapting to easily. Some of the parameters which affect the complexity of the harbor environment are:

- . The number of vessels which traverse the harbor daily;
- . The waterway geography;
- . The types and numbers of sensors available;
- . The class of the system;
- . The current mode of operation.

Five levels of VTS systems have been defined based on the type of sensor which provides vessel position and course information. These sensor levels are:

- . Level 1 - Location reports by voice on radio;
- . Level 2 - Manual or automatic point sensors (e.g., magnetic or acoustic sensors which indicate a vessel's passage without identifying it);
- . Level 3 - Manual area sensors, which may detect position and course information automatically, but require manual entry of this information into the processing/display subsystem;

- . Level 4 - Automatic tracking by radar and input of position and course information;
- . Level 5 - Automatic tracking using active ship-board electronics such as radar transponders, or Loran-C retransmitters.

The VTS processing/display subsystem is also described by its class and mode. The three possible modes are:

- . Mode A - Informing, for which the following types of information are provided;
 - Traffic information
 - . Identity of nearby vessels and buoys
 - . Predictions of future encounters
 - . Prediction of traffic congestion
 - Navigational Information
 - . Unusual weather conditions
 - . Tide and current conditions
 - . Status of aids-to-navigation
 - . Hazards to navigation
 - . Maritime events of particular concern
- . Mode B - Hazard Detection and Reporting, which provides, in addition to Mode A services, detection of the following types of hazards, which may be accomplished automatically, depending on the availability of appropriate sensors;
 - Grounding
 - Congestion

- Lane Stray
- Route Stray
- Dangerous Encounter
- Collision
- Excessive Vessel Speed
- Navaid Adrift/Missing
- Anchor Drift

- . Mode C - Routing, which provides, in addition to the Mode A and B services, congestion free vessel routing through the waterway.

The class of a system refers to its maximum operating capability. The class of a system is designated using letters which specify the highest operating modes. A Class B system may operate in either Mode A or B. A Class B system, however, cannot operate in Mode C because the capability would not exist in a Class B system.

Most combinations of Class and Level are possible, depending on the harbor parameters and the relative need for accident prevention. A Class A system, however, can not support level 4 or 5 sensors.

2.5 VTS SYSTEM DESIGN GOALS

Several design goals have been established for the VTS system so that a single system design can be used to provide the class and level of service required at any given port or waterway. The hardware and software must have sufficient capability and flexibility to meet all the varying needs imposed by the different classes and levels.

Some specific design goals or requirements are:

- . Modularity - which allows adaptability of the system to a wide variety of harbors, expandability (and contractability) to handle increased (decreased) numbers of vessels, higher (lower) sensor levels, and greater (less) coverage area. Modularity also improves maintainability by making trouble shooting easier. Modularity also allows the standardization of hardware and software which facilitates training of hardware maintenance crews and watchstanders;
- . Distributed Processing - using multiple processors to share the processing load and provide a high degree of reliability (i.e., 99.9% availability is required);
- . Off-the-shelf hardware, to the extent that it is consistent with the other goals;
- . The VTS processing/display subsystem must accommodate a maximum of:
 - one watch supervisor station

- ten traffic coordinator stations
- three external communicator stations
- one spare station which can be dedicated to training when all stations are operational

2.6 VTS SYSTEM DATA BASE

The VTS System Data Base will consist of five major logical files. These files will vary in size depending on the individual requirements of a particular VTS System. In the following brief descriptions the number of records required for the maximum configuration is given.

- . Vessel File - contains background information regarding vessels which traverse the VTS coverage areas frequently. This eliminates collecting data repetitively each time a vessel makes a passage through the VTS coverage area. In the maximum configuration the vessel file will have a capacity of 10,000 vessels.
- . Passage File - contains information for each vessel while it traverses the VTS coverage area. This information includes the vessel's identification code, cargo, points of entry into and expected exit from the VTS coverage area, as well as intended route. Passage status, reflecting the state of the vessel in its passage, may be:
 - imminent - vessel about to enter the coverage area (i.e., begin its passage)
 - underway - passage currently in progress
 - docked or anchored - passage ended within the coverage area

The passage file has a capacity of 2,000 passages in the maximum configuration.

- . Waterway File - contains information about normal and special routes through the harbor, map cell data (e.g., depths), waypoints (which facilitate position reporting in a Level 1 system), docks, piers and Nav aids.
- . Environment File - contains weather, water current and tide information collected from manual or automatic environment sensors in the VTS coverage area. Its capacity is 20 automatic weather sensors, 20 automatic current/tide sensors, and 40 inputs from manual sensors of either type.
- . Notices File - contains the text of official notices pertaining to some or all of the VTS coverage area. Its capacity is 100 notices.

For additional details on the data base contents and required access rate, see Section 4.4.

2.7 PROCESSING CAPABILITIES

The capabilities of the VTS Processing Display Subsystem for processing input data fall into four categories:

- . Background Processing
- . File Operations
- . Demand Processing
- . Support Functions

Background processing consists of hazard detection, position update and system logging functions. File operations processing includes various data base functions for building the data files, modifying them, and deleting selected information. Demand processing allows the analysis and display of additional information in response to requests from the watchstander. Support functions consist of utility and exception functions designed to aid system usefulness and enhance system management and reliability. For further details on processing capabilities see Section 3.

3.1 INTRODUCTION

Processing may be defined as the manipulation of input data and the preparation of output data. It is those operations which transform input into output.

To determine processing requirements, we will separate both input and output operations from pure processing. We will assume that, processing of input begins when all the input data is available in memory, and output commences when all output data has been prepared and stored in one or more output memory buffers. The time required to read from or write to peripheral devices will be excluded from consideration here. Also, the time required for communicating input or output data between processors will be excluded from consideration. However, the time required for I/O and interprocessor communications will be discussed later, since it is in some respects dependent upon the architecture under consideration.

VTS processing may be separated into four categories:

- . Background Processing
- . File Operations
- . Demand Processing
- . Support Functions

Background processing is defined as the execution of those applications functions which are regularly scheduled by the operating system. Background processing consists mainly of hazard detection

functions, and excludes some other regularly scheduled functions, such as hardware diagnosis and failure detection, which are categorized as support functions (discussed below).

File operations processing is defined as the execution of those applications functions necessary to maintain the data base, excluding the time required to read from or write to various logical files in the data base.

Demand processing is defined as the execution of those applications functions, executed only as necessary, which are designed to assist the watchstander and/or supervisor in the performance of their jobs. Demand processing consists mainly of functions which provide additional information about vessel characteristics or alert situations.

Support functions processing is defined as the execution of those functions which are required for system management purposes. It includes such functions as simulation, map generation, and error recovery.

All of the processing categories are defined and discussed in more detail in subsequent paragraphs.

Of these four categories, only the first does not normally require manual initiation or intervention. For the other three categories, generally some manual input will be required to initiate the various functions in response to an external stimulus, and in some cases these functions require more processing per iteration than background processing. Thus, it is expected that the background processing functions will cause a relatively higher average processing load than other functions, while the file operations, demand processing and support functions will cause peaking effects in the processing.

3.2 APPROACH

A two step approach will be used to estimate VTS processing requirements. For a first approximation, conservative estimates will be made of the processing requirements. For a second approximation, we will examine more closely those functions which make a relatively large contribution to the processing load. We will refine the estimates for these functions to reduce their processing load contributions where possible.

For each approximation we will examine the effect of the harbor environment by considering three different harbor scenarios. These scenarios represent a set of assumptions for use in evaluation. Actual configurations would be based on the actual needs of the particular harbor and would be expected to differ from these.

- . A Class C, Level 4 system with 900 identified vessels simultaneously in transit. This is chosen to represent a worst case traffic management environment, for maximum design purposes (Scenario 1). A total of 12 display stations would be included. Eight tracking radars have also been assumed.
- . A Class B, Level 4 system with 150 identified and 350 unidentified vessels simultaneously in transit. This is chosen to represent the maximum requirements of a typical harbor environment (Scenario 2). Seven display stations and three tracking radars have been assumed.
- . A Class A, Level 1 system with 100 identified vessels simultaneously in transit. This is chosen to represent the maximum requirements of a smaller harbor environment (Scenario 3). No radars are included in a Class A, Level 1 system. Five display stations are assumed.

To represent processing requirements quantitatively, we will calculate the number of machine cycles per second required for each function. We will estimate the number of iterations of each function required per second, and multiply by the estimated number of machine cycles required for each iteration of each function. Summing over all functions will yield the total estimated VTS processing load in terms of machine cycles per second.

For the background processing, we will calculate the average processing load imposed by assuming each function is executed at its maximum specified frequency. For the other three categories we will estimate the peak processing load required. Depending on the data available, we will make two types of assumptions to estimate peak processing requirements:

- . In cases where we can estimate the average processing load, (e.g., by knowledge of the total number of iterations required per day) we will assume that the peak load is equal to four times the average;
- . In cases where average processing load data is not available, we will make an assumption about the peak frequency of execution, based on how the function is used and the harbor scenario under consideration.

Various assumptions will be made to estimate the number of machine cycles required per iteration. These assumptions will be discussed as appropriate in the analysis given below for each function. However, in cases where instruction estimates are made, we make the following two assumptions:

- . An average of two machine cycles per instruction is required. This is typical of today's mini-computers;
- . A safety factor of 2.5 is applied to the estimate of machine cycles. This is done to allow for variations in the number of machine cycles per instruction, and to allow for the possibility that not all necessary instructions have been identified (e.g., the precise number of instructions required for a function may depend heavily on the hardware and software architecture chosen).

Thus, we will multiply the estimated number of instructions per iteration by five to obtain the estimated number of machine cycles per iteration. (For the second approximation we will depart from this method in selected cases.)

Floating point hardware will not be assumed in the first approximation since floating point hardware is not available for all classes of processors which may be considered for the VTS system. Floating point hardware will be considered as a part of the second approximation.

3.3 PROCESSING CATEGORIES

This section describes in greater detail the functions associated with each processing category.

The following define the functions and associated maximum frequency of execution for background processing.

<u>Function</u>	<u>Frequency (seconds)</u>
. Hazard detection	see below
. Vessel position update	6
. System backup data update	60
. Vessel passage history update	variable
. VTS operations log update	variable

Hazard detection, vessel position update, and system backup data update are regularly scheduled for execution, although the last of these three may be executed relatively slowly in comparison (i.e., an execution frequency of up to ten minutes). The last two of the above categories are event driven, and thus may provide a smaller contribution to the processing load than some of the others.

Detection of vessel hazards is done with nine different functions. These functions, and their maximum frequency of execution are:

<u>Function</u>	<u>Frequency (seconds)</u>
. Potential collisions	
Stage 1	30
Stage 2	15
Stage 3	6
. Lane stray	30
. Route stray	30

<u>Functions</u>	<u>Frequency (seconds)</u>
. Potential grounding	30
. Excessive congestion	30
. Dangerous encounter	30
. Anchor drift	60
. Navaid adrift or missing	60
. Excessive vessel speed	60

File Operations processing consists of functions designed to maintain each of the files in the data base. For each file there are the three basic functions, enter, modify and delete. For the passage file, several other functions must be implemented. The complete list of applications files, functions, and average number of executions per passage as follows:

<u>File/Function</u>	<u>Executions per passage</u>
. Vessel File	
- Enter new vessel	.2
- Modify vessel data	.02
- Delete vessel	.2
. Passage File	
- Enter new passage	.6
- Identify vessel	.9
- Update vessel position	8.0
- Modify passage information	.2
- Enter new communications	5.0
- Delete passage	.6
- Change status	1.2

<u>File/Function</u>	<u>Executions per passage</u>
. Waterway File	N/A
- Enter waterway data	
- Modify waterway data	
- Delete waterway data	
. Environment File	N/A
- Enter environment data	
- Modify environment data	
- Delete environment data	
. Notices File	N/A
- Enter notice	
- Modify notice	
- Delete notice	

Demand processing involves 6 major functions:

- . Identify encounters
- . Determine relative position between two vessels
or a vessel and a point
- . Determine closest point of approach (CPA)
- . Schedule/route vessel traffic
- . Search the data base for selected or predefined
data
- . Respond to an alert

The data base searches are further subdivided:

- . Identify all vessels within a given area around a point (local traffic)
- . Display environmental information (weather or current and tide)
- . Display notices information
- . Search the data base for data defined by a key consisting of up to 10 parameters
- . Display all information in the data base for a particular
 - vessel or
 - passage

Support functions consist of utility and exception functions. The utility functions are:

- . Simulation of a waterway environment
- . Testing of new software
- . Implementation of a new data base
- . Program (software) development
- . Map generation

Simulation enables watchstander training in an artificial environment consisting of up to 100 vessels. It consists of execution of all background file and demand functions which are applicable for the particular system under consideration.

Exception functions are:

- . Failure (hardware and software) detection
- . Error recovery
- . Hardware diagnosis
- . Power failure recovery

3.4 FIRST APPROXIMATION

For the First Approximation we will determine quantitative processing requirements for the three harbor scenarios defined above. The three scenarios will then be summarized for each processing category.

3.4.1 Scenario 1

The first scenario involves a Class C, Level 4 system with 900 identified vessels in transit. Subsequent paragraphs discuss the processing requirements for each category.

3.4.1.1 Background Processing

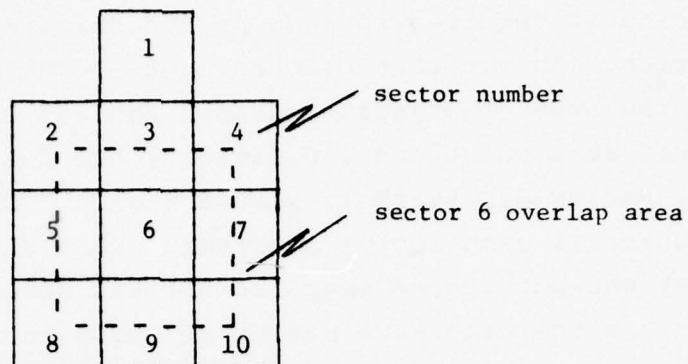
Figure 3-1 indicates, for each of the background functions, the estimated number of iterations per second, number of machine cycles per iteration, and the total number of machine cycles per second. The method and assumptions used to arrive at these numbers are described below.

Potential Collisions

For a waterway with n ships, all of which are identified, the number of pairs of vessels which must be examined for stage 1 of potential collision processing is $n(n-1)/2$. Since the number of calculations to be performed varies roughly as $n^2/2$, a method of reducing the effective n to be considered is desirable. This may be done by subdividing the waterway. A possible sectorization is shown in Figure 3-2. This arrangement of sectors was chosen to maximize the number of boundaries between sectors. Also, the number of vessels in each sector is uniform, with the exception of the innermost sector, number 6, which is assumed to experience the greatest activity; i.e., it is assumed that there will be one sector of each waterway which has more activity in

<u>Process Type</u>	<u>Number of Iterations Per Second</u>	<u>Number of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Potential Collisions			
Stage 1	13,269	510	6,767,190
Stage 2	6	300	1,800
Stage 3	7.5	505	3,788
Lane Stray	30	10,000	300,000
Route Stray	30	10,000	300,000
Grounding	30	26,000	780,000
Dangerous Encounter	30	500	15,000
Excessive Congestion	30	2,000	60,000
Anchor Drift	15	300	4,500
Excessive Speed	15	300	4,500
Navaid Adrift/Missing	33	300	9,900
Position Update	150	10,000	1,500,000
System Backup Data Update	15	250	3,750
Vessel Passage History Update	10	1,000	10,000
VTS Operations Log Update	10	1,000	<u>10,000</u>
TOTAL			9,770,428

Figure 3-1. Estimated VTS Background Processing Load For A
Class C, Level 4 System With 900
Identified Vessels (First Approximation)



<u>Sector Number</u>	<u>Nominal Number of Vessels</u>	<u>Total Number of Vessels (incl. overlap)</u>	<u>Number of Iterations per Second</u>
1	80	160	424
2	80	230	878
3	80	340	1,921
4	80	230	878
5	80	300	1,495
6	200	440	3,219
7	80	300	1,495
8	80	210	732
9	80	300	1,495
10	80	210	732
TOTALS	920*	2,720	13,269

*A slightly greater number of vessels is assumed here for computational convenience.

Figure 3-2. Potential Collisions - Stage 1 - Number Of Iterations Per Second (First Approximation)

terms of vessel passages (i.e., 200 compared to 80), than any other sector. In addition, assume that some overlap in sector monitoring is required to account for ships at or near sector boundaries. Assume that processing for each sector must include 50% of the area of adjacent sectors and 25% of sectors which intersect at a point, as indicated by the dotted line in Figure 3-2 for sector 6. Further, assume that all ships are uniformly distributed in each sector quadrant. Then the total number of calculations per second required for all sectors is $\sum n_i(n_i-1)/60$ where n_i is the effective number of ships in the i th sector, and the divisor of 60 takes into consideration the fact that this function need not be done faster than once every 30 seconds.

It should be noted that sectorization of this type does not result in a major reduction in the processing required for collision detection. While there is a net reduction in the number of pairs to be considered, this reduction is not as substantial as might have been anticipated because of the combined effects of the worst case assumptions which have been made.

An actual waterway could normally be divided into sectors in such a way that processing would be substantially reduced. Indeed it might be true that no harbor which would utilize this system would approach the worst case model during the life of the system. Since such assumptions cannot be justified without extensive study of traffic patterns in all prospective VTS sites, the worst case assumptions will be used for this first approximation of collision processing.

The estimated number of instructions executed per iteration in Stage 1 is 102, assuming no vessels are exempt, vessels are sufficiently close to require a distance calculation, and no

other stage of collision processing is in effect. For Stage 2 of collision processing, it is assumed that 10% of the vessels have another vessel close enough to require this level of processing every 15 seconds. In this case, time to CPA and distance from CPA are calculated. Approximately 60 instructions per iteration are assumed to be executed.

For Stage 3 of collision processing, assume that half the vessels in Stage 2 require this level of processing every 6 seconds. Here a relatively complex risk calculation is necessary, requiring 4 table lookups for each pair along with either the issue, update or cancellation of an alert (i.e., we assume at least one of these happens each iteration). An estimated 101 instructions per iteration are required for Stage 3 collision processing.

Lane Stray

For this case the overriding assumption is that both a sine and cosine calculation are required, each requiring approximately 5000 cycles. Other processing is assumed to be negligible in comparison. In addition, all vessels are assumed to be in lanes for this process.

Route Stray

The same analysis as for Lane Stray applies here.

Potential Grounding

For this calculation, assume that all vessels in the environment require a maximum number of cells to be examined. Further assume that 160 cells are the maximum number to be examined in 40 steps; (i.e., 40 cells can be traversed in 10 minutes at 30 mph, and at each cell, 4 cells in the direction of travel must be examined to determine which one of these cells the vessel will next pass through. Also, assume that 20 instructions are required to identify each cell and 50 instructions are required at each of 40 steps to complete the grounding calculation.

Dangerous Encounters

In this process we assume that all vessels are examined every 30 seconds to determine whether they are about to enter a constricted area or are about to cross a channel. If either situation is true, further checking will be performed to determine if another vessel will be encountered and if so, whether the encounter can be handled safely. Approximately 100 instructions per iteration will be required for each vessel.

Excessive Congestion

The assumption of major significance here is that floating point calculations are required, with an average of 2,000 cycles per iteration per vessel, at a rate of 30 iterations per second (i.e., all vessels are considered).

Anchor Drift

For purposes of conservatively estimating this processing, assume all vessels are at anchor. A calculation of distance to anchorage position will be required, with an estimated 60 instructions per vessel.

Excessive Speed

Assume that the current position cell is checked for each vessel to determine whether a speed limit is in effect, requiring 60 instructions each time.

Navaid Adrift/Missing

Assume 2,000 nav aids are examined every 60 seconds. The calculation is essentially the same as "Anchor Drift" above, i.e., 60 instructions per nav aid.

Vessel Position Update

For this case we assume that all vessel positions are updated each 6 seconds. The calculation requires:

- . An examination of boundary conditions, such as a vessel entering, leaving or disappearing temporarily from sensor tracking;
- . A coordinate transformation (from polar to Cartesian coordinates);
- . A coordinate origin translation (from the radar origin to the system origin);
- . Locating and updating a vessel's coordinates, velocity components and corresponding time of measurement.

The coordinate transformation is of major significance, requiring a calculation of both the sine and cosine of the bearing angle, each of which consumes approximately 5,000 machine cycles per iteration. The other processing is assumed to be negligible in comparison.

System Backup Data Update

In the worst case, backup data must be prepared every 60 seconds, for each vessel in passage. Approximately ten data elements per vessel must be prepared and recorded. Assuming five instructions per data element, 50 instructions are necessary for each vessel each iteration. Also, assuming the waterway is sectorized, no sort by sector of backup data will be necessary.

Vessel Passage History Update

Assuming an average of two status changes are required for each vessel which traverses a waterway, 1,800 passage history updates per day will be required. However, since a manual status change is a prerequisite for a passage history entry, assume that each watchstander makes a status change at a peak rate of once per second. Then 10 entries per second must be made. Assume an average of 20 data elements per entry and 10 instructions per data element, or 200 instructions per entry.

VTs Operations Log Update

Assume that for each sector a log entry is required at a peak rate of once per second, or 10 calculations per second. Also, assume that the log entry is approximately the same size and complexity as the passage history entry. Then 200 instructions per entry are required.

3.4.1.2 File Operations Processing

Figure 3-3 summarizes the processing load for file operations. Using the execution frequencies per passage given above for the Vessel and Passage files, and multiplying by 900 passages yields the average number of iterations per second required. Peak rates, for design purposes, are assumed to be four times the average rate.

The Waterway and Notices files are relatively static. We assume that these two files produce a negligible contribution to the processing load. For the environment file, we assume an average execution frequency of one of the three basic file operations (enter, modify, delete) per second.

For the Vessel and Passage files we assume that 500 instructions are required per disc access (the number of disc accesses per function is given in Section 4.4). Multiplying by the safety factor of five yields the number of machine cycles per iteration for the Vessel and Passage files. For the Environment file, we assume an average of 200 instructions are required for each iteration.

3.4.1.3 Demand Processing

To arrive at a peak processing load for demand functions, we assume a peak execution frequency of one per second for each of the following functions: (See Figure 3-4)

- . Dangerous encounters
- . Relative position
- . CPA

<u>Process Type</u>	<u>Peak Number of Iterations Per Second</u>	<u>Number of Machine Cycles Per Iteration*</u>	<u>Number of Machine Cycles Per Second</u>
Vessel File			
Enter New Vessel	.008	50,000	400
Modify Vessel Data	.0008	12,500	10
Delete Vessel	.008	62,500	500
Passage File			
Enter New Passage	.024	75,000	1,800
Identify Vessel	.036	27,500	990
Update Vessel Position	.32	12,500	4,000
Modify Passage Information	.008	12,500	100
Enter New Communication	.20	20,000	4,000
Delete Passage	.024	57,500	1,380
Change Status	.048	12,500	600
Waterway File	~0		~0
Environment File	4	1,000	4,000
Notices File	~0		~0
TOTAL			17,780

*500 instructions per disc access is assumed.

Figure 3-3. Estimated VTS Peak File Operations Processing Load For A Class C, Level 4 System With 900 Identified Vessels (First Approximation)

<u>Process Type</u>	<u>Number of Iterations Per Second</u>	<u>Number of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Dangerous Encounters	1	510	510
Relative Position	1	15,000	15,000
CPA	1	300	300
Schedule/Route Traffic			375,000
Data Base Information Requests			
Local Traffic	440*	510	224,400
Environmental Information	1	5,000	5,000
Notices	1	10,000	10,000
Search on a Key	40	2,500	100,000
Predefined Searches			
Display Vessel Information			
Display Passage Information			
Alert Response	1	20,000	20,000
TOTAL			750,210

*worst case sector

Figure 3-4. Estimated VTS Peak Demand Processing Load
for a Class C, Level 4 System with 900
Identified Vessels. (First Approximation)

- . Environmental information
- . Notices
- . Alert response

For the local traffic search, we assume that it is done on a sector basis and that all vessels in the sector must be examined as potential candidates for inclusion in the output. For the worst case sector, containing 440 vessels, as indicated in Figure 3-2, we assume 440 iterations per second are required. This is done in effect to establish a response time of approximately one second for this function.

For the search on an operator-defined key and the two predefined searches, we assume that forty vessels per second can be examined. We also assume that only one of the three searches is in progress at any given time.

To estimate the number of machine cycles per iteration, we assume the following for the effort required for the demand functions:

- . Dangerous encounters - requires approximately the same effort as Stage 1 of collision processing;
- . Relative position - requires a range and bearing calculation. The range computation requires a square root function, and the bearing calculation requires an inverse trigonometric function. We assume a total of 15,000 cycles for these two operations, and that other processing is negligible in comparison;
- . CPA - requires approximately the same processing as potential collision processing, Stage 2;

- . Local traffic - requires approximately the same processing as Stage 1 of potential collision processing;
- . Environmental information - requires 1000 instructions per iteration;
- . Notices - requires 2000 instructions per iteration
- . Searches - requires 500 instructions per vessel;
- . Alert response - requires an average of 4000 instructions to handle each watchstander request during the response to an alert (note that the watchstander may make several requests for information about the alert).

Scheduling/routing of vessels is applicable only for a Class C system. To determine processing requirements, we make the following assumptions:

- . Each of the 900 vessels is scheduled/routed daily
- . Three routes are examined for each vessel
- . Thus 2700 routes/schedules must be determined daily, or at an average of approximately one every thirty seconds
- . Fifty route segments must be examined for each routing/scheduling problem (i.e., for each vessel)
- . 250 machine cycles are required for processing each route segment

- . Thus 11,250,000 machine cycles per iteration are required

This processing assumes that route and/or schedule conflicts between vessels can be identified. However, we assume that no elaborate optimization techniques (such as linear programming) are used in the analysis. Also, we assume that, in cases where determining a route and schedule is difficult, either the watchstander and pilot will resolve the situation cooperatively, or the machine processing involved to resolve the difficulty occurs rarely enough to be considered negligible.

3.4.1.4 Support Function Processing

Figure 3-5 summarizes the processing load for support functions in this scenario. For the utility functions, we assume that the greatest processing load is caused by execution of the simulation function. Simulation, when executed, produces an estimated processing load given in Figure 3-6. Note that in Figure 3-6, we assume only one waterway sector, and 100 vessels instantaneously in transit. In addition, we have estimated a 50,000 cycle per second contribution for the file and demand functions.

We assume that only one of the utility functions is executed at any given time, and that none of the utility functions produces a greater peak processing load than simulation.

For the exception functions we estimate that the total peak processing load caused by all these functions is 100,000 cycles per second. It is difficult to make more than a rough estimate of the processing load caused by the exception functions at this point, because the actual processing load may depend heavily on the hardware and software architecture chosen. In addition, arbitrary decisions may be made later which influence this processing (e.g., a requirement that all hardware diagnostic functions be executed once per minute).

<u>Process Type</u>	<u>Number of Machine Cycles Per Second</u>
Utility Functions	493,167
- Simulation	
- Software Testing	
- Data Base Implementation	
- Program Development	
- Map Generation	
Exception Functions	100,000*
- Failure Detection	
- Error Recovery	
- Hardware Diagnosis	
- Power Failure Recovery	
TOTAL	593,167

*Estimated

Figure 3-5. Estimated VTS Peak Support Functions
Processing Load For A Class C, Level 4
System With 900 Identified Vessels (First Approximation)

<u>Process Type</u>	<u>Number of Iterations Per Second</u>	<u>Number of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Potential Collisions			
Stage 1	165	510	84,150
Stage 2	.67	300	201
Stage 3	.833	505	421
Lane Stray	3.3	10,000	33,000
Route Stray	3.3	10,000	33,000
Grounding	3.3	26,000	85,800
Dangerous Encounter	3.3	500	1,650
Excessive Congestion	3.3	2,000	6,600
Anchor Drift	1.7	300	510
Excessive Speed	1.7	300	510
Navaid Adrift/Missing	33	300	9,900
Position Update	16.7	10,000	167,000
System Backup Data Update	1.7	250	425
Vessel Passage History Update	10	1,000	10,000
VTIS Operations Log Update	10	1,000	10,000
File and Demand Functions			50,000*
TOTAL			493,167

*Estimated

Figure 3-6. Estimated VTS Peak Simulation Processing Load for a Class C, Level 4 System

3.4.2 Scenario 2

The second scenario involves a Class B, Level 4 system with 150 identified vessels and 350 unidentified vessels.

For Scenario 2, we eliminate the assumption of a ten sector waterway. We now assume a one sector waterway. The only other basic change in this scenario involves the number of iterations per second for each function, which must be changed because it is a function of the number of vessels simultaneously in transit. The number of machine cycles per iteration is the same as in Scenario 1. Therefore, in the subsequent paragraphs for this scenario, we will discuss the method used to arrive at the number of iterations per second.

3.4.2.1 Background Processing

For the following functions, the number of iterations per second is not different from Scenario 1: (See Figure 3-7)

- . Navaid Adrift/Missing - which is a function of the number of nav aids (2000) assumed
- . Vessel Passage History Update
- . VTS Operations Log Update

The last two of the above functions are assumed not to be a function of the number of vessels in the scenario.

For the other functions, we use the same maximum execution frequency as in Scenario 1. However, for the following functions, analysis is necessary only for the 150 identified vessels: (i.e., the unidentified vessels are not considered)

- . Lane Stray
- . Route Stray

<u>Process Type</u>	<u>Number of Iterations Per Second</u>	<u>Number of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Potential Collisions			
Stage 1	2,123	510	1,082,730
Stage 2	3	300	900
Stage 3	4	505	2,020
Lane Stray	5	10,000	50,000
Route Stray	5	10,000	50,000
Grounding	5	26,000	130,000
Dangerous Encounter	5	500	2,500
Excessive Congestion	5	2,000	10,000
Anchor Drift	3	300	900
Excessive Speed	3	300	900
Navaid Adrift/Missing	33	300	9,900
Position Update	83	10,000	830,000
System Backup Data Update	3	250	750
Vessel Passage History	10	1,000	10,000
VTS Operations Log Update	10	1,000	10,000
TOTAL			2,190,600

Figure 3-7. Estimated VTS Background Processing Load For A
Class B, Level 4 System with 150 Identified
Vessels and 350 Unidentified Vessels (First Approximation)

- . Grounding
- . Excessive Congestion
- . Anchor Drift
- . System Backup Data Update

For potential collisions processing, Stage 1, we must analyze all pairs of vessels except those for which both vessels are unidentified.

In general, the number of vessel pairs to be examined, $f(n)$, is equal to $n(n-1)/2$ where n is the number of vessels. The number of vessel pairs to be examined in this case is $f(500)-f(350)$. Dividing by the maximum execution frequency of 30 seconds, yields the number of iterations per second, 2123. For Stages 2 and 3 we make the same assumptions as in Scenario 1.

For the position update function, we must update the position of all 500 vessels in the scenario every six seconds, or approximately 83 times per second.

3.4.2.2 File Operations Processing

For the Vessel and Passage files, we use the same method as in Scenario 1. We multiply the number of identified vessels by the execution frequency per passage to arrive at the average number of iterations per second. Again the processing load for the Waterway and Notices Files are assumed to be negligible.

For the Environment file, we assume a peak number of iterations of one per minute or approximately .017 per second.

For the Vessel and Passage files, again the peak processing load is equal to four times the average. (see Figure 3-8)

<u>Process Type</u>	<u>Peak Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Vessel File			
Enter New Vessel	.0012	50,000	60
Modify Vessel Data	.00012	12,500	2
Delete Vessel	.0012	62,500	75
Passage File			
Enter New Passage	.004	75,000	300
Identify Vessel	.008	27,500	220
Update Vessel Position	.056	12,500	700
Modify Passage Information	.004	12,500	50
Enter New Communication	.036	20,000	720
Delete Passage	.004	57,500	300
Change Status	.008	12,500	100
Waterway File	~ 0		0
Environment File	.017*	1,000	17
Notices File	~ 0		0
TOTAL			2,544

*Peak rate of 1 per minute

Figure 3-8. Estimated VTS Peak File Operations Processing
Load For A Class B, Level 4 System With 150
Identified And 350 Unidentified Vessels (First Approximation)

3.4.2.3 Demand Processing

The method used here is the same as for Scenario 1 with the following exceptions: (see Figure 3-9)

- . Scheduling/routing is not applicable in a Class B system;
- . The number of iterations per second for Local Traffic is 500, the same as the total number of vessels.

3.4.2.4 Support Functions Processing

The processing load for Scenario 2 is the same as Scenario 1 for the support functions. Simulation processing does not change, nor does the exception function processing. The peak number of machine cycles per second is assumed to be the same as for Scenario 1.

3.4.3 Scenario 3

The third scenario involves a Class A, Level 1 system with 100 identified vessels simultaneously in transit. Again for this scenario no changes to the number of machine cycles per iteration are assumed. The basic change from Scenario 2 is to the number of iterations per second for most functions. These changes reflect the fact that some functions are not applicable for a Class A, Level 1 System, as discussed below.

3.4.3.1 Background Processing

The following hazard detection functions are not applicable in a Class A, Level 1 system:

- . Potential Collisions (all stages)
- . Lane Stray
- . Grounding
- . Anchor Drift
- . Navaid Adrift/Missing

<u>Process Type</u>	<u>Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Dangerous Encounters	1	510	510
Relative Position	1	15,000	15,000
CPA	1	300	300
Schedule/Route Traffic	N/A		0
Data Base Information Requests			
Local Traffic	150*	510	76,500
Environmental Information	1	5,000	5,000
Notices	1	10,000	10,000
Searches	40	2,500	100,000
Alert Response	1	20,000	20,000
TOTAL			227,310

*Worst Case Sector

Figure 3-9. Estimated VTS Peak Demand Processing Load For
A Class B, Level 4 System With 150
Identified, 350 Unidentified Vessels (First Approximation)

For Route Stray, the execution frequency is the same as for position update (which is essentially a manual operation in a Level 1 system). We assume a peak position update frequency of four times the average rate of 800 per day (for a 100 vessel environment). The same method as for Scenario 2 applies for the other background functions. (See Figure 3-10)

3.4.3.2 File Operations Processing

The number of iterations per second for the Vessel and Passage file functions is less here than for the other two scenarios because it is a function of the number of vessels. The same method as for Scenario 2 is used here. A peak rate of one per minute is assumed for the environment file, and the waterway and notices files are again assumed to produce a negligible contribution to the processing load. (See Figure 3-11)

3.4.3.3 Demand Processing

For the demand functions, we make the following changes to the assumptions for the number of iterations per second: (See Figure 3-12)

- . We assume a peak rate of one per minute for Dangerous Encounters, Relative Position, CPA, Environmental Information, and Notices;
- . Scheduling/routing is not applicable (requires a Class C system);
- . For Local Traffic, we assume, as before, a number equal to the number of vessels in the scenario.

<u>Process Type</u>	<u>Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Potential Collisions	N/A		0
Lane Stray	N/A		0
Route Stray	.036*	10,000	360
Grounding	N/A		0
Dangerous Encounter	3	500	1,500
Excessive Congestion	3	2,000	6,000
Anchor Drift	N/A		0
Excessive Speed	3	300	900
Navaid Adrift/Missing	N/A		0
Position Update	.036**	10,000	360
System Backup Data Update	2	250	500
Vessel Passage History Update	10	1,000	10,000
VTS Operations Log Update	10	1,000	10,000
TOTAL			29,620

*Same as position update frequency

**Four times average rate of 800 per day.

Figure 3-10. Estimated VTS Background Processing Load For A
Class A, Level 1 System With 100
Identified Vessels (First Approximation)

<u>Process Type</u>	<u>Peak Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Vessel File			
Enter New Vessel	.0008	50,000	40
Modify Vessel Data	.00008	12,500	1
Delete Vessel	.0008	62,500	54
Passage File			
Enter New Passage	.0024	75,000	180
Identify Vessel	.004	27,500	110
Update Vessel Position	.036	12,500	450
Modify Passage Information	.0008	12,500	10
Enter New Communication	.024	20,000	480
Delete Passage	.0024	57,500	138
Change Status	.0048	12,500	60
Waterway File	~ 0		0
Environment File	.017*	1,000	17
Notices File	~ 0		0
TOTAL			1,540

*Peak rate of 1 per minute

Figure 3-11. Estimated VTS Peak File Operations Processing Load For A Class A, Level 1 System With 100 Identified Vessels (First Approximation)

<u>Process Type</u>	<u>Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Demand Functions			
Dangerous Encounters	.017*	510	9
Relative Position	.017*	15,000	255
CPA	.017*	300	5
Schedule/Route Traffic	N/A		
Data Base Information Requests			
Local Traffic	100**	510	51,000
Environmental Information	.017*	5,000	85
Notices	.017*	10,000	170
Searches	40	2,500	100,000
Alert Response	1	20,000	20,000
TOTAL			171,524

*Peak rate of 1 per minute assumed

**Worst case

Figure 3-12. Estimated VTS Peak Demand Functions Processing Load
For A Class A, Level 1 System With 100
Identified Vessels (First Approximation)

For the data base searches and the alert response, no changes are made.

3.4.3.4 Support Functions Processing

The support functions processing load changes for Scenario 3 because the simulation processing load changes. This is because we assume that the simulation function can only simulate those functions which are applicable to the Class and Level of the system. Thus the following functions, and associated processing loads, are eliminated for this scenario: (see Figure 3-13 and 3-14)

- . Potential Collisions (all stages)
- . Lane Stray
- . Grounding
- . Anchor Drift
- . Navaid Adrift/Missing

Other simulation function processing load contributions remain the same as for Scenario 2.

3.4.4 Summary of the Results of the First Approximation

Figure 3-15 presents a summary, for each of the three scenarios, of the processing load contribution for each processing category. The total peak period processing loads are:

- . Scenario 1 - 11,131,585 cycles/second
- . Scenario 2 - 3,192,021 cycles/second
- . Scenario 3 - 382,304 cycles/second

<u>Process Type</u>	<u>Number of Machine Cycles Per Second</u>
Utility Functions	79,620
Simulation	
Software Testing	
Data Base Implementation	
Program Development	
Map Generation	
Exception Functions	100,000*
Failure Detection	
Error Recovery	
Hardware Diagnosis	
Power Failure Recovery	
TOTAL	179,620

*Estimated

Figure 3-13. Estimated VTS Peak Support Functions
Processing Load For A Class A, Level 1 System
With 100 Identified Vessels (First Approximation)

<u>Process Type</u>	<u>Number Of Iterations Per Second</u>	<u>Number Of Machine Cycles Per Iteration</u>	<u>Number of Machine Cycles Per Second</u>
Potential Collisions	N/A		0
Lane Stray	N/A		0
Route Stray	.036*	10,000	360
Grounding	N/A		0
Dangerous Encounter	3	500	1,500
Excessive Congestion	3	2,000	6,000
Anchor Drift	N/A		0
Excessive Speed	3	300	900
Navaid Adrift/Missing	N/A		0
Position Update	.036**	10,000	360
System Backup Data Update	2	250	500
Vessel Passage History Update	10	1,000	10,000
VTS Operations Log Update	10	1,000	10,000
File and Demand Operations			50,000 ⁺
TOTAL			79,620

*Same as position update frequency

**Four times average rate of 800 per day

⁺Estimated

Figure 3-14. Estimated VTS Peak Simulation Processing Load
For A Class A, Level 1 System
(First Approximation)

--- NUMBER OF MACHINE CYCLES PER SECOND --- (First Approximation)			
Processing Category	900 Vessels Class C Level 4	500 Vessels ⁺ Class B Level 4	100 Vessels Class A Level 1
Background	9,770,428	2,190,600	29,620
File Operations	17,780	2,544	1,540
Demand	750,210	227,310	171,524
Support	593,167	593,167	179,620
TOTAL	11,131,585	3,013,621	382,304

⁺150 identified vessels, 350 unidentified vessels.

Figure 3-15. Summary of Estimated VTS Peak Processing Loads In
Three Different Harbor Environments
(First Approximation)

3.5 SECOND APPROXIMATION

A review of the processing requirements we have developed as a first approximation shows three functions which contribute significantly to the overall processing load. To arrive at a second approximation, we will concentrate on techniques that can be used to reduce the processing load from these three functions.

The largest single processing load results from the first stage of collision processing. In Scenario 1 the first stage of collision processing accounts for more than 60% of the total processing load. The next largest processing load is the position update function which contributes over 13% of the total load. The other function contributing a substantial load is the grounding function at 5%. Together these three functions account for over 80% of the total.

It will be necessary to review all of the functions at some point in the design of the VTS Processing/Display Subsystem. Significant changes may be made in the estimates for any of the functions as the design progresses. It should be clear, however, that even a 100% change in the processing load estimates for one of the other functions would have relatively little impact on the total load estimate. A similar percentage change in the estimate for the first stage of collision processing would, however, have a dramatic impact on the overall estimate.

First Stage of Collision Processing

The first stage of collision processing is amenable to a change in algorithm to simplify the processing and to the efficient coding of the central loop. The algorithm assumed in the first approximation included a calculation of the square of the distance between vessel pairs. The original approach also assumed that vessels were grouped by sectors which introduces a degree of overhead and, as previously noted, does not result in a substantial savings when the necessary worst case is considered.

For the second approximation, sectorizing is no longer considered and the algorithm has been altered to examine individual coordinate differences rather than the square of the distance. This makes it possible to write a one page assembly language routine to scan all possible vessel pairs comparing both X and Y coordinates. Based on the actual machine cycles required on two different minicomputers, 36 machine cycles will be required per iteration.

The same algorithm can be applied to the local traffic function. Both functions assume the new algorithm for the second approximation.

Position Update

The processing load for position update function is primarily the result of sine and cosine calculations. Floating point hardware is the most effective way to reduce this processing load since floating point hardware typically reduces the processing for calculations of this type by a factor of ten. Floating point hardware would also reduce the processing load estimates for the following functions:

- Lane Stray
- Route Stray
- Relative Position
- Excessive Congestion

Floating point hardware, however, may not be available for all classes of processors which are appropriate to consider for the VTS Processing/Display subsystem. For the second approximation, systems with and without floating point hardware will be considered and two sets of estimates prepared.

Grounding

For grounding, we find from a review of the assumptions made for the first approximation that the worst case basis used was unrealistic since the speed assumed for vessels was 30 mph. While some vessels might travel at that speed in a VTS area, the average speed of the vessels which should be used in determining the processing load would not be that high. A more reasonable (still conservative) estimate of the average speed would be 12 mph. which results in 32 cells to be examined in 8 steps. The number of instructions required for each cell (20) and step (50) will remain the same as for the first approximation.

The following estimates are based on these modified assumptions for collision detection, position update, and grounding as well as the other functions which are also affected by the changed algorithms or the inclusion of floating point hardware. This second approximation should provide a more realistic model of the processing required for the VTS Processing/Display Subsystem.

3.5.1 Scenario 1

Again we assume a Class C, Level 4 system with 900 identified vessels instantaneously in transit. The additional two columns in all the figures reflect the effect of assuming floating point hardware.

3.5.1.1 Background Processing

The number of iterations per second for potential collision processing, Stage 1 is equal to $f(900)/30$ where $f(n)$ is equal to $\frac{n(n-1)}{2}$. A summary of the processing load, reflecting the other assumptions mentioned above, is presented in Figure 3-16.

3.5.1.2 File Operations Processing

The processing load contribution for file operations is the same as for Scenario 1 presented for the first approximation. (See Figure 3-3)

Process Type	----- N u m b e r O f -----				
	Iterations Per Second	Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Potential Collisions					
Stage 1	13,485	36	36	485,460	485,460
Stage 2	6	300	300	1,800	1,800
Stage 3	7.5	505	505	3,788	3,788
Lane Stray	30	10,000	1,000	300,000	30,000
Route Stray	30	10,000	1,000	300,000	30,000
Grounding	30	5,200	5,200	156,000	156,000
Dangerous Encounter	30	500	500	15,000	15,000
Excessive Congestion	30	2,000	200	60,000	6,000
Anchor Drift	15	300	300	4,500	4,500
Excessive Speed	15	300	300	4,500	4,500
Navaid Adrift/ Missing	33	300	300	9,900	9,900
Position Update	150	10,000	1,000	1,500,000	150,000
System Backup Data Update	15	250	250	3,750	3,750
Vessel Passage History Update	10	1,000	1,000	10,000	10,000
VTS Operation Log Update	10	1,000	1,000	10,000	10,000
TOTAL				2,864,698	920,698

*Without floating point hardware.

**With floating point hardware.

Figure 3-16. Estimated VTS Background Processing Load
for a Class C, Level 4 System with
900 Identified Vessels (Second Approximation)

3.5.1.3 Demand Processing

This is the same as for Scenario 1, First Approximation, except for the following: (See Figure 3-17)

- . The relative position function is affected by floating point hardware;
- . The local traffic function reflects the fact that no waterway sectorization is assumed. Thus all 900 vessels must be examined each time the function is used. A more efficient algorithm based on the approach used for collision detection can be used, however.

3.5.1.4 Support Functions Processing

The changes to the support function processing load, Figure 3-18, reflect the assumptions given above which affect the background processing portion of the simulation load. We assume that the file and demand functions processing load does not change, as well as the exception function processing load. See Figure 3-19 for the simulation processing load.

3.5.2 Scenario 2

Scenario 2 again involves a Class B, Level 4 system with 150 identified vessels and 350 unidentified vessels instantaneously in transit.

3.5.2.1 Background Processing

The only changes from the First Approximation are to reflect the basic assumption changes noted above. The result is shown in Figure 3-20.

3.5.2.2 File Operations Processing

The processing load for file operations is identical to that in Scenario 2, First Approximation. (See Figure 3-8)

Process Type	Iterations Per Second	N u m b e r O f			
		Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Dangerous Encounters	1	500	500	500	500
Relative Position	1	15,000	1,500	15,000	1,500
CPA	1	300	300	300	300
Schedule/Route Traffic				375,000	375,000
Data Base Information Requests					
Local Traffic	900	36	36	32,400	32,400
Environmental Information	1	5,000	5,000	5,000	5,000
Notices	1	10,000	10,000	10,000	10,000
Searches	40	2,500	2,500	100,000	100,000
Alert Response	1	20,000	20,000	20,000	20,000
TOTAL				558,200	544,700

*Without floating point hardware.
**With floating point hardware.

Figure 3-17. Estimated VTS Peak Demand Processing Load
for a Class C Level 4 System with 900 Identified Vessels
(Second Approximation)

Process Type	Number of Machine Cycles Per Second	
	*	**
Utility Functions	346,317	130,677
Simulation		
Software Testing		
Data Base Implementation		
Program Development		
Map Generation		
Exception Functions	100,000 ⁺	100,000 ⁺
Failure Detection		
Error Recovery		
Hardware Diagnosis		
Power Failure Recovery		
TOTAL	446,317	230,677

*Without floating point hardware

**With floating point hardware.

⁺Estimated

Figure 3-18. Estimated VTS Peak Support Functions
Processing Load For A Class C, Level 4
System With 900 Identified Vessels (Second Approximation)

Process Type	Iterations Per Second	Number Of			
		Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Potential Collisions					
Stage 1	165	36	36	5,940	5,940
Stage 2	.67	300	300	201	201
Stage 3	.833	505	505	421	421
Lane Stray	3.3	10,000	1,000	33,000	3,300
Route Stray	3.3	10,000	1,000	33,000	3,300
Grounding	3.3	5,200	5,200	17,160	17,160
Dangerous Encounter	3.3	500	500	1,650	1,650
Excessive Congestion	3.3	2,000	200	6,600	660
Anchor Drift	1.7	300	300	510	510
Excessive Speed	1.7	300	300	510	510
Navaid Adrift/ Missing	33	300	300	9,900	9,900
Position Update	16.7	10,000	1,000	167,000	16,700
System Backup Data Update	1.7	250	250	425	425
Vessel Passage History Update	10	1,000	1,000	10,000	10,000
VTS Operations Log Update	10	1,000	1,000	10,000	10,000
File and Demand Functions				50,000	50,000
TOTAL				346,317	130,677

*Without floating point hardware.

**With floating point hardware.

Figure 3-19. Estimated VTS Peak Simulation Processing Load
for a Class C, Level 4 System (Second Approximation)

Process Type	Iterations Per Second	----- N u m b e r O f -----			
		Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Potential Collisions					
Stage 1	2,123	36	36	76,428	76,428
Stage 2	3	300	300	900	900
Stage 3	4	505	505	2,020	2,020
Lane Stray	5	10,000	1,000	50,000	5,000
Route Stray	5	10,000	1,000	50,000	5,000
Grounding	5	5,200	5,200	26,000	26,000
Dangerous Encounter	5	500	500	2,500	2,500
Excessive Congestion	5	2,000	200	10,000	1,000
Anchor Drift	3	300	300	900	900
Excessive Speed	3	300	300	900	900
Navaid Adrift/ Missing	33	300	300	9,900	9,900
Position Update	83	10,000	1,000	830,000	83,000
System Backup Data Update	3	250	250	750	750
Vessel Passage History Update	10	1,000	1,000	10,000	10,000
VTS Operation Log Update	10	1,000	1,000	10,000	10,000
TOTAL				1,080,298	234,298

*Without floating point hardware.

**With floating point hardware.

Figure 3-20. Estimated VTS Background Processing Load
for a Class B, Level 4 System with 150
Identified 350 Unidentified Vessels (Second Approximation)

3.5.2.3 Demand Processing

Changes to demand processing reflect the assumption changes mentioned above. The resulting peak demand processing load is presented in Figure 3-21.

3.5.2.4 Support Functions Processing

This processing load is identical to Scenario 1, Second Approximation. (See Figure 3-18).

3.5.3 Scenario 3

Scenario 3 involves a Class A, Level 1 system, and as before, the major differences from the other two scenarios are caused by the elimination of selected functions because of the absence of sensor support for them.

3.5.3.1 Background Processing

Reflecting the effect of eliminating some functions, and the basic changes defined for the Second Approximation, yields the background processing load given in Figure 3-22.

3.5.3.2 File Operations Processing

This processing load is identical to that in Scenario 3, First Approximation (Figure 3-11).

3.5.3.3 Demand Processing

The reduction in processing load compared to Scenario 3, First Approximation is caused by the changes to the assumptions as defined above. The result is given in Figure 3-23.

3.5.3.4 Support Functions Processing

The support function processing load is reduced by reflecting the changes, caused by the assumptions for the second approximation, to the simulation function. The support functions processing load is presented in Figure 3-24, while the simulation processing load is shown in Figure 3-25. Note that the processing load caused by the exception functions does not change.

Process Type	Iterations Per Second	N u m b e r O f			
		Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Dangerous Encounters	1	500	500	500	500
Relative Position	1	15,000	1,500	15,000	1,500
CPA	1	300	300	300	300
Schedule/Route Traffic	N/A			0	0
Data Base Information Requests					
Local Traffic	150	36	36	5,400	5,400
Environmental Information	1	5,000	5,000	5,000	5,000
Notices	1	10,000	10,000	10,000	10,000
Searches	40	2,500	2,500	100,000	100,000
Alert Response	1	20,000	20,000	20,000	20,000
TOTAL				156,200	142,700

*Without floating point hardware.
**With floating point hardware.

Figure 3-21. Estimated VTS Peak Demand Processing
Load for a Class B, Level 4 System with
150 Identified, 350 Unidentified Vessels (Second Approximation)

Process Type	----- N u m b e r O f -----				
	Iterations Per Second	Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Potential Collisions	N/A			0	0
Stage 1					
Stage 2					
Stage 3					
Lane Stray	N/A			0	0
Route Stray	.036	10,000	1,000	360	36
Grounding	N/A			0	0
Dangerous Encounter	3	500	500	1,500	1,500
Excessive Congestion	3	2,000	200	6,000	600
Anchor Drift	N/A			0	0
Excessive Speed	3	300	300	900	900
Navaid Adrift/ Missing	N/A			0	0
Position Update	.036	10,000	1,000	360	36
System Backup Data Update	2	250	250	500	500
Vessel Passage History Update	10	1,000	1,000	10,000	10,000
VTS Operation Log Update	10	1,000	1,000	10,000	10,000
TOTAL				29,620	23,572

*Without floating point hardware.

**With floating point hardware.

Figure 3-22. Estimated VTS Background Processing Load
for a Class A, Level 1 System with 100
Identified Vessels (Second Approximation)

Process Type	Iterations Per Second	N u m b e r O f			
		Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Dangerous Encounters	.017	500	500	9	9
Relative Position	.017	15,000	1,500	255	26
CPA	.017	300	300	5	5
Schedule/Route Traffic	N/A			0	0
Data Base Information Requests					
Local Traffic	100	36	36	3,600	3,600
Environmental Information	.017	5,000	5,000	85	85
Notices	.017	10,000	10,000	170	170
Searches	40	2,500	2,500	100,000	100,000
Alert Response	1	20,000	20,000	20,000	20,000
TOTAL				124,124	123,895

*Without floating point hardware.

**With floating point hardware.

Figure 3-23 Estimated VTS Peak Demand Processing Load
a Class A, Level 1 System with 100
Identified Vessels (Second Approximation)

Process Type	Number of Machine Cycles Per Second	
	*	**
Utility Functions	79,620	73,572
Simulation		
Software Testing		
Data Base Implementation		
Program Development		
Map Generation		
Exception Functions	100,000 ⁺	100,000 ⁺
Failure Detection		
Error Recovery		
Hardware Diagnosis		
Power Failure Recovery		
TOTAL	179,620	173,572

*Without floating point hardware

**With floating point hardware.

⁺Estimated

Figure 3-24. Estimated VTS Peak Support Functions
Processing Load For A Class A, Level 1
System With 100 Identified Vessels (Second Approximation)

Process Type	----- N u m b e r O f -----				
	Iterations Per Second	Machine Cycles Per Iteration		Machine Cycles Per Second	
		*	**	*	**
Potential Collisions	N/A			0	0
Stage 1					
Stage 2					
Stage 3					
Lane Stray	N/A			0	0
Route Stray	.036	10,000	1,000	360	36
Grounding	N/A			0	0
Dangerous Encounter	3	500	500	1,500	1,500
Excessive Congestion	3	2,000	200	6,000	600
Anchor Drift	N/A			0	0
Excessive Speed	3	300	300	900	900
Navaid Adrift/ Missing	N/A			0	0
Position Update	.036	10,000	1,000	360	36
System Backup Data Update	2	250	250	500	500
Vessel Passage History Update	10	1,000	1,000	10,000	10,000
VTS Operations Log Update	10	1,000	1,000	10,000	10,000
File and Demand Functions				50,000	50,000
TOTAL				79,620	73,572

*Without floating point hardware.

**With floating point hardware.

Figure 3-25. Estimated VTS Peak Simulation Processing Load
for a Class A, Level 1 System (Second Approximation)

3.5.3.5 Summary of the Results of the Second Approximation
Figure 3-26 presents, for each scenario, the processing loads for each processing category. The minimum processing loads for each scenario, i.e., assuming the presence of floating point hardware, are as follows:

- . Scenario 1 - 1,707,995 cycles/second
- . Scenario 2 - 610,219 cycles/second
- . Scenario 3 - 322,579 cycles/second

----- NUMBER OF MACHINE CYCLES PER SECOND ----- (Second Approximation)						
Processing Category	900 Vessels		500 Vessels ⁺		100 Vessels	
	Class C *	Level 4 **	Class B *	Level 4 **	Class A *	Level 1 **
Background	2,858,698	914,838	1,080,298	234,298	29,620	23,572
File Operations	17,780	17,780	2,544	2,544	1,540	1,540
Demand	558,200	544,700	156,200	142,700	124,124	123,895
Support	446,317	230,677	446,317	230,677	179,620	173,572
TOTAL	3,880,995	1,707,995	1,685,359	610,219	334,904	322,579

⁺150 Identified vessels, 350 unidentified vessels

*Without floating point hardware.

**With floating point hardware.

Figure 3-26. Summary of Estimated VTS Peak Processing Loads In
Three Different Harbor Environments
(Second Approximation)

3.6 SUMMARY

A comparison of the results of the First Approximation with those of the Second Approximation shows a dramatic decrease in the processing load estimate. This was achieved by concentrating on the few major contributors to processing load.

Two conclusions can be reached from an analysis of the two results:

- . As in many real time systems, concern for efficiency in a very small percentage of the software can result in a substantial reduction in the total processing load. It is not necessary to tightly code the entire system to achieve most of the possible benefits of tight coding. This is encouraging, since the USCG desires a modular, highly structured approach to software.
- . For the larger configurations (Scenarios 1 and 2), floating point hardware results in a substantial reduction in the processing load estimate. Since floating point hardware is inexpensive relative to computers, floating point hardware should be assumed for the larger configurations if it is available for the processors being considered.

4.1 INTRODUCTION

In this section we will discuss the mass storage files which are required to implement the VTS Processing/Display Subsystem according to the functional requirements specified by the USCG.

The data base will be designed to emphasize both retrieval simplicity and efficiency because of the real-time nature of the system and because of the response time requirements set forth in the functional specification. We do not foresee using a complete data base management system (DBMS) because of the processing overhead associated with separating the physical and logical data files. Instead, we propose making the logical and physical files identical, and designing data access methods appropriate for each file, taking into consideration the number of keys utilized, as well as response time requirements.

Maximum retrieval efficiency is not sought. The goal is a design which will conservatively meet the response time requirements (e.g., for the worst case data access, the response time will be less than or equal to the required response time).

In the discussion which follows, the information content and field and record sizes are presented for the major system files. At this early stage of the design these file descriptions are presented as a basis for estimating storage capacities and defining the data base structure. As the design progresses, file and record structures will be refined and the data base designed to provide the flexibility to change the types of data and the record and field sizes. This flexibility will be needed to allow the basic VTS system to adapt easily to new and different requirements which will almost certainly be encountered as VTS is implemented for different ports and harbors.

4.2 ACCESS METHODS

In this section we will discuss the ways which will be used to access the physical data base. The data access method for indexed files will be different from that used for non-indexed files.

For files which require one or more indexes, we will use a multi-level index structure. Figure 4-1 shows the index organization for these types of files. At each level of index, the keys will be in ascending alphanumeric order. At all levels except the lowest level, the entries will indicate the first key in that block of keys. By comparing successive pairs of entries at the first level, the block containing the next (or last) index level can be identified. For all levels except the lowest, each entry will contain the key and a pointer to the beginning of the next block of keys. At the lowest level, the index entry will contain the key and the location of the desired record. An example of a three level index structure will illustrate the concept.

Suppose we have a vessel file consisting of 1000 records, and we want to set up an index structure using the (alphabetic) vessel name as the key. An example of the method which could be used to set up a three level index is as follows:

- . The vessel names would be arranged in alphabetic order. The vessels could then be divided into ten groups of 100. Ten index entries could be set up using the key (and pointer to the next level) for the first vessel in each group. Determining that a given vessel name lies alphabetically between key n and key $n+1$ would narrow down the next block examined to block n .

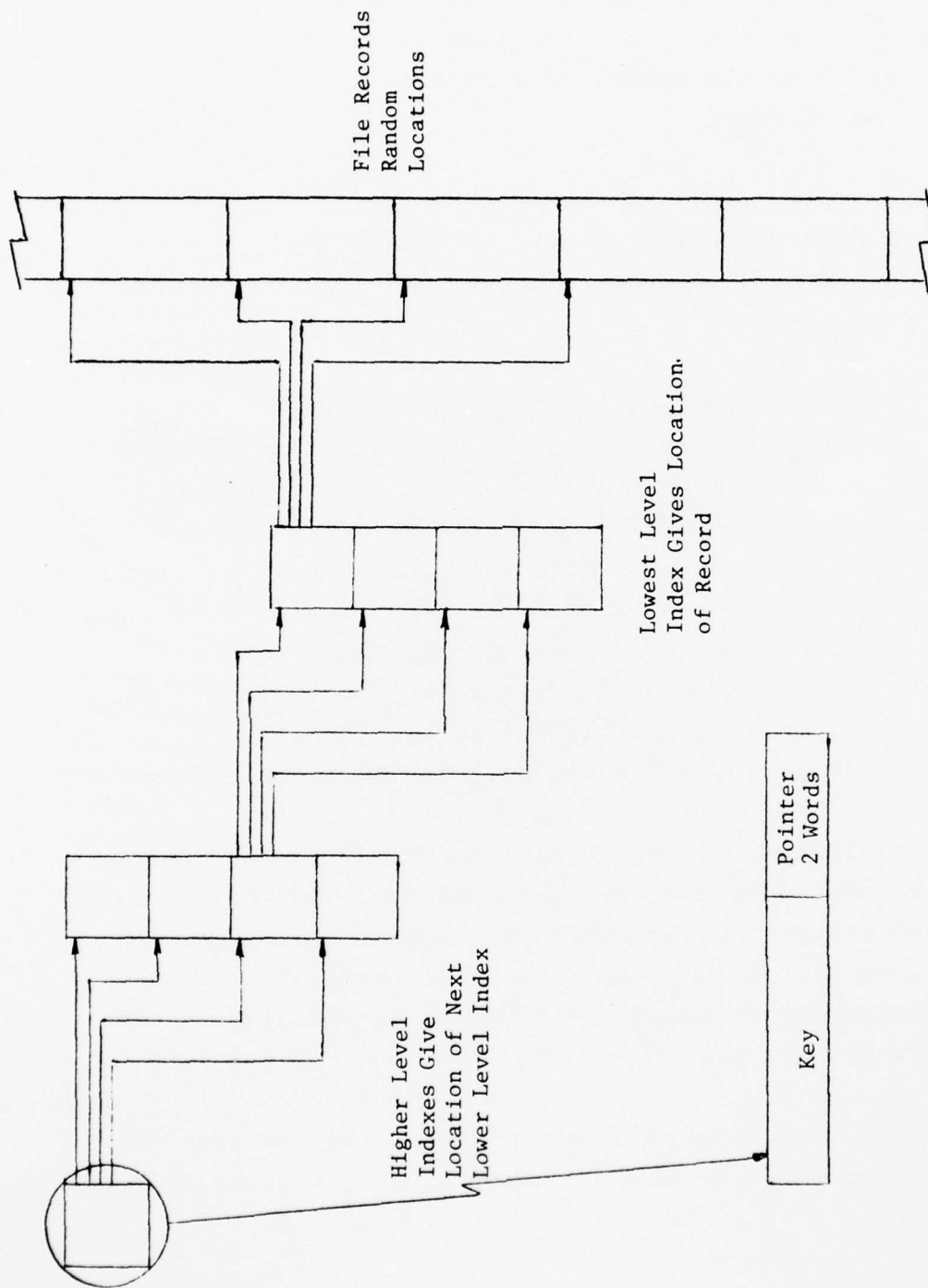


Figure 4-1. Multi-Level Index Structure

- . At the second index level, the groups of 100 could be divided into blocks of 10, with each of the ten entries set up and searched as above. This reduces to 10 the number of remaining vessels to be examined.
- . At the third index level, up to 10 keys are examined to find the vessel name. At that point the entry contains a pointer to the actual vessel record desired.

Care must be taken when creating or eliminating a vessel record. When creating a vessel record, modification of all three indexes could be necessary if the entry requires addition to a block which is full. This is because we are assuming a fixed maximum for the number of entries in each block, so that the index blocks do not become disproportionate in size, thereby causing potentially drastic differences in response times for data access, as a function of the location of an entry within an index block. When deleting a vessel, similar modifications to the indexes are required if the deleted vessel causes an index block to be emptied.

For access to non-indexed files, or for searches on a key or keys which are not supported by indexes, the entire subject file will be examined to find the particular record or records of interest or to satisfy the requirements of the particular search (i.e., to produce the desired output parameters subject to the limitations imposed by the key or keys).

These access methods provide the basis on which the analysis of disc accessing frequency was done. This analysis is given in Section 4.4.

4.3 FILE STRUCTURES AND SIZES

The data required for the VTS application may be divided into the following categories:

- . Vessel data - provides background information on each vessel which transits the waterway frequently.
- . Passage data - provides rapidly changing information for each vessel while it is in passage, i.e., within the VTS coverage area.
- . Waterway data - provides physical waterway information necessary to perform hazard detection.
- . Environment data - provides information about the weather in the waterway, as well as current and tide data.
- . Notices data - provides information regarding official notices pertaining to parts or all of the waterway.

The information required to describe the waterway and environment has been separated into several files. For the waterway data, the following files have been defined to allow easier access to the data:

- . Route segments
- . Waterway cells (primary and supplementary)
- . Waypoints
- . Docks and piers
- . Nav aids

Likewise, for the environment data, the following files have been identified:

- . Weather sensors
- . Current and tide sensors
- . Manual weather data
- . Forecasts

For the passage file, we have separated the identification and text of all communications into a separate file, because of the potential length of this data.

Figure 4-2 summarizes the mass storage requirements for the applications data files. This figure indicates the maximum storage needed for each file. The maximum number of records, the record size, and the total file size is given for all file categories except three, the map storage, software and miscellaneous files. For these three file categories only an estimate of required storage can be made until details of the software structure are developed.

The file sizing data given in Figure 4-2 includes only the logical storage requirements for each file. When the logical file requirements are mapped onto some physical disc device, there will be some unused space. This unused space results from taking into consideration the physical disc sector size. For example, if a logical record in a given file is 250 bytes long, and the disc sector size is 256 bytes, it is typical to assign one record per sector, leaving six bytes per sector for this file unused. Multiple records per sector may be assigned if the sector size is greater than or equal to an integral multiple of the logical record size. Thus, the number of unused bytes is a function of the disc sector size, and so we will not estimate the extra space required, at this point.

FILE	NUMBER OF RECORDS	RECORD SIZE(BYTES)	FILE SIZE(BYTES)
Vessel File	10,000	369	3,690,000
Vessel Indexes	10,000	72	720,000
Passage File	2,000	424	848,000
Passage Indexes	2,000	66	132,000
Communications	40,000	168	6,720,000
Route Segments	220	96	21,120
Waterway Cells			
- Primary	160,000*	16	2,560,000
- Supplementary	16,000	294	4,704,000
Waypoints	400	30	12,000
Docks and Piers	600	118	70,800
Nav aids	2,000	14	28,000
Weather Sensors	20	25	500
Current and Tide Sensors	20	22	440
Manual Weather Data	40	219	8,760
Forecasts	10	210	2,100
Notices	100	10,087	1,008,700
Map Storage			10,000,000
Software Files			20,000,000
Miscellaneous			4,000,000
TOTAL			54,526,420

*Maximum total coverage

Figure 4-2. File Sizing Summary

4.3.1 Vessel File

Figure 4-3 presents vessel file sizing data. For each data element the units (if any), and the number of bytes required is given. Implicit in the number of bytes per entry for each data element is the requirement that it is large enough to accommodate the longest possible entry especially for those elements with varying lengths.

The vessel file has four different keys through which background information about any vessel may be accessed or defined:

- . Lloyd's Registry Number (hull number for military vessels)
- . Radio call sign
- . Name
- . Location

The location key is the only one not stored in the file, and is primarily used for determining the identity of a vessel in passage, based on an operator supplied location.

4.3.2 Passage File

Figure 4-4 provides sizing information for the passage file. This file contains basic information for the passage, as well as other time dependent information for each vessel depending on its passage status (for underway, anchored or docked). If the passage status is imminent, only the basic data (see Figure 4-4) is necessary. As noted above, information about communications has been defined as a separate file.

In Level 4 and 5 systems, position, course, speed and size data about each vessel in passage are not stored in the passage file.

DATA ELEMENT	UNITS	LENGTH (BYTES)
Vessel Name	Characters	25
Lloyd's Registry or Military Hull No.	Characters	8
Radio Call Sign	Characters	6
Type	Types	1
Gross Weight	Tons	4
Flag	Characters	14
Owner	Characters	36
Maximum Speed	Knots	1
Maximum Draft	Feet	1
Minimum Draft	Feet	1
Beam	Feet	2
Overall Length	Feet	2
Overall Height at Minimum Draft	Feet	1
Doctor Aboard Normally	Yes - No	1
Local Agent - Name	Characters	36
Address & Phone #	Characters	55
Time Required for Crash Stop	Seconds	2
Type of Navigational Equipment	Types	2
Number of Screws	---	1
Minimum Turning Radius	Feet	2
Date Data Last Verified	--	2
Date Vessel Last Active	--	2
Miscellaneous Comments	Characters	160
Linkage Pointer	--	4
TOTAL		369

Figure 4-3. Vessel File Sizing

DATA ELEMENT	UNITS	LENGTH (BYTES)
<u>Basic Data</u>		
Vessel ID Code	Characters	4
Vessel Name	Characters	25
Passage Status	States	1
Present Sector	Sectors	1
Origin or Point of Entry to VTS	Coordinates	4
	Characters	40
Destination or Point of Exit	Coordinates	4
	Characters	40
Date/Time of Entry	Coordinates	4
Scheduled Date/Time of Exit		4
Pilot - Designation	Characters	6
Name	Characters	36
Barge Makeup	Characters	40
Cargo	Types	1
Actual Draft	Feet	1
Intended Route	Segments	100
Link to Vessel File		4
Link to Communications		4
TOTAL		319
<u>Underway (Levels 1, 2, 3)</u>		
For Each Checkpoint Passed (Up to 50):		
Checkpoint Designation	Characters	4
Date/Time at Checkpoint		4
Speed of Advance to Next Checkpoint	Knots	1
Designation of Next Checkpoint	Characters	4
Time of Arrival at Next Checkpoint		2
TOTAL		15
<u>Anchored (All Levels)</u>		
Anchorage Designation	Characters	14
Swing Radius of Mooring	Feet	2
Location of Anchorage	Coordinates	4
	Characters	40
Date/Time Anchorage Established		4
Date/Time Expected to Get Underway		4
TOTAL		68
<u>Docked (All Levels)</u>		
Dock or Pier Designation	Characters	14
Date/Time Arrived		4
Date/Time Scheduled to Depart		4
TOTAL		22
<u>Communications</u>		
Link Pointer		4
Date/Time of Communication		4
Summary of Communication		160
TOTAL	Characters	168

Figure 4-4. Passage File Sizing

Instead, these data will be kept in a memory table, because they change rapidly, and are used extensively for hazard detection calculations. For unidentified vessels only this type of data is stored.

The passage file data may be accessed through five keys:

- . Vessel ID code
- . Name
- . Pilot Designation
- . Tracker ID (assigned by the tracking system in Level 4 and 5 systems)
- . Location

The tracker ID and location keys are not kept in the passage file, but will be used internally to identify vessels. In addition, the location key may be used by the watchstander to identify a vessel in passage.

4.3.3 Waterway Characteristics File

Figure 4-5 gives the file size for the waterway file. Excluded from this figure is information regarding the geographical breakdown of the waterway into map cells. This is treated in section 4.3.4.

The Waterway file is subdivided functionally into four files as mentioned above. Relatively static information is contained in each of these four files describing normal and special routes taken by vessels in passage, waypoints used to estimate vessel

position, the various docks and piers in the waterway, and aids to navigation present in the waterway.

4.3.4 Waterway Cell Data Base Structure

The VTS coverage area will be subdivided into cells which will be used to describe the characteristics of the waterway. The waterway database is structured at three levels, the highest of which is resident in memory. See Figure 4-6.

At the first level the VTS area is divided into cell blocks. Each block represents a 10 x 10 matrix of cells. The memory resident descriptor is a single word containing:

- . Mean lowest low water (MLLW)
- . Flags describing features of the cell block including:
 - Routes/lanes
 - Docks/piers
 - Hazards
 - Navigational aids
 - Sensors
 - Waypoints

In a maximum coverage VTS, the memory table would represent a 40 x 40 matrix with each word describing a block of cells with a total size of approximately 10 nautical miles on a side.

Each cell block is a 10 x 10 matrix of individual cell descriptions which are each 8 words. Each cell represents an area that is approximately 1 nautical mile on a side. Included in the cell descriptor is the following: (See Figure 4-7)

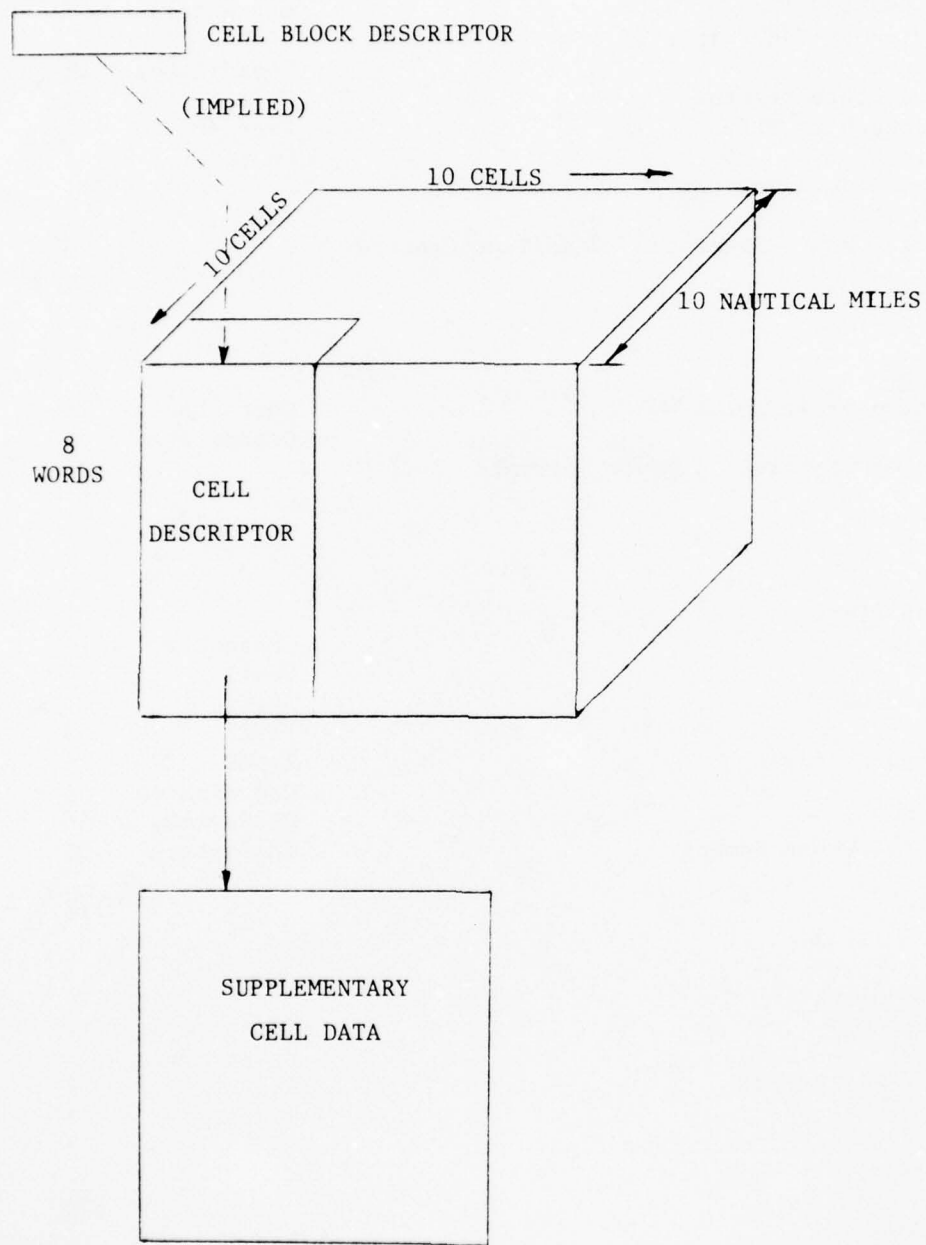


Figure 4-6. Waterway Data Base Description

<u>DATA ELEMENT</u>	<u>UNITS</u>	<u>LENGTH (BYTES)</u>
<u>Route Segments</u>		
Segment Designation	Characters	6
Coordinates of End Points of Straight Line Portions	Coordinates	80
Length of Route Segment	Feet	4
Minimum Depth at MLLW	Feet	1
Speed Limit	Knots	1
For Normal Route Segment:* Minimum Width of Channel	Feet	2
For Special Route Segment:* Date/Time Created		4
TOTAL		94 - 96
<u>Waypoints</u>		
Designation of Waypoint	Characters	6
Location	Coordinates	4
Location of Waypoint in Route Segments		20
TOTAL		30
<u>Docks and Piers</u>		
Designation	Characters	14
Length	Feet	2
Depth at MLLW	Feet	1
Width	Feet	2
Facilities Available	Types	4
Location	Coordinates	4
Name of Owner	Characters	36
Address and Phone Number	Characters	55
TOTAL		118
<u>Navigational Aids</u>		
Designation	Characters	6
Location	Coordinates	4
Watch Circle Radius	Feet	2
Type of Aid	Types	1
Adrift or Missing Processing Flag		1
TOTAL		14

*Only one of these two entries is required.

Figure 4-5. Waterway Characteristics File Sizing

<u>CELL DESCRIPTION</u>	<u>LENGTH (BYTES)</u>
MLLW	1
Speed Limit	1
Flags	2
Pointer to Supplemental Info	4
Subcell MLLW's 4 x 4 x 1/2	8
	<hr/>
TOTAL	16

<u>SUPPLEMENTARY DATA</u>	
Hazard Codes	2
Detailed MLLW's 16 x 16	256
Notice, Text 36 Characters	36
	<hr/>
TOTAL	294

Figure 4-7. Waterway Cell Data

AD-A061 307

INTERNATIONAL COMPUTING CO BETHESDA MD

F/G 15/5

PRELIMINARY DESIGN STUDY FOR VTS PROCESSING/DISPLAY SUBSYSTEM.(U)

JUN 78 C C HENSON, R A CLEAVER, S H KAISLER

DOT-CG-81-78-1833

USCG-D-66-78

NL

UNCLASSIFIED

2 OF 4

AD
A061307



. Mean lowest low water for the cell	1 byte
. Speed limit in cell	1 byte
. Flags describing cell features	2 bytes
. Subcell descriptor	8 bytes
. Pointer to supplementary data	4 bytes

The subcell descriptor included above contains a 4 x 4 matrix of 4 bits each. The data items are a coded representation of the MLLW within a cell. The value zero indicates a water level equal to the MLLW for the entire cell. Values 1 through 15 determine the water level by the formula:

$$MLLW_{\text{subcell}} = MLLW_{\text{cell}} + 2 \times \text{VALUE}$$

The cell descriptor contains the MLLW for each area of the waterway down to approximately 1/4 nautical mile on a side. If additional detail is needed for MLLWs or if the flag word indicates the presence of hazards or other features requiring elaboration, a detail block will exist containing the required data.

4.3.5 Environment File

Figure 4-8 gives the size of the environment file, which has also been subdivided into four files. These files contain information, as a function of location within the waterway, about the

- . weather status, such as temperature, visibility, from automatic and manual input;
- . waterway status, such as direction and speed of current, and tide level;
- . weather forecast.

<u>DATA ELEMENT</u>	<u>UNITS</u>	<u>LENGTH (BYTES)</u>
<u>Automatic Weather Stations</u>		
Designation	Characters	14
Location	Coordinates	4
Temperature	Degrees	1
Visibility	Miles	1
Precipitation Rate	Scaled	2
Wind Direction	Degrees	2
Wind Speed		1
TOTAL		<u>25</u>
<u>Automatic Current and Tide Sensors</u>		
Designation	Characters	14
Location	Coordinates	4
Direction of Current	Degrees	2
Speed of Current		1
Tide Level Referred to MLLW		1
TOTAL		<u>22</u>
<u>Manual Data Inputs</u>		
Source of Data	Characters	36
Date/Time Entered		4
Location Where Valid	Sector(s)	1
	Coordinates	4
Key Words	Characters	14
Free Form Text	Characters	160
TOTAL		<u>219</u>
<u>Forecasts</u>		
Source	Characters	36
Date/Time Entered		4
Date/Time Span Valid		8
Area Covered	Sector(s)	2
Forecast	Characters	160
TOTAL		<u>210</u>

Figure 4-8. Environment File Sizing

4.3.6 Notices File

Figure 4-9 lists the information necessary for the notices file. The main portion of this file is the text of the notice. Other information identifies and describes the notice, in terms of its applicability to a waterway area (i.e., sector(s)) and the time period for which the notice is valid.

4.3.7 Map Display Storage

Ten basic maps (maximum) are required for the VTS coverage area (one per sector). We assume 10,000 vectors for each basic map (x1 magnification). In addition, three levels of magnification are required, x2, x3, x5. We assume that the number of vectors required for the magnifications is equal to 10,000 multiplied by the square of the magnification factor. Thus, for all four magnification levels, 390,000 vectors are required for each map. Since at this level of detail most vectors would be represented by short vector forms requiring 2 bytes, we will allow approximately 10 million bytes of storage for map data.

4.3.8 Software Files

Storage will be necessary for software object files in all systems, and for source files in systems which have the program development capability. This storage must include sufficient space for applications and operating system software necessary for all processors. The source and object data will not be stored in the same file. In addition, applications and operating systems software will require separate files. Finally, the software necessary for each processor may be grouped and stored with the processor, depending on the architecture chosen.

For a first approximation of the total software storage required, we have estimated 20 million bytes, based on experience with similar systems.

<u>DATA ELEMENT</u>	<u>UNITS</u>	<u>LENGTH (BYTES)</u>
Type of Notice	Types	1
Light List Number if Applicable	Characters	36
Portion(s) Covered by Notice	Sectors	2
Name or Number of Navaid if Applicable	Characters	36
Date/Time Entered		4
Date/Time Span Valid		8
Text	Characters	10,000
TOTAL		<u>10,087</u>

Figure 4-9. Notices File Sizing

4.3.9 Miscellaneous Files

The files described so far do not necessarily define precisely the total data base required. Various other data files may be required, as a function of the architecture chosen, to manage interprocessor communications, set up and schedule processor tasks, or provide temporary storage for intermediate results of a procedure. We estimate that roughly 4 million bytes could be required for these types of files, again distributed (or not) as necessary among various physical devices.

4.4 DISC ACCESS FREQUENCIES

In this section we will develop the number of disc accesses required for the major watchstander functions in the three scenarios defined in Section 3:

- . Scenario 1 - A Class C, Level 4 system with 900 passages per day for identified vessels
- . Scenario 2 - A Class B, Level 4 system with 150 passages per day for identified vessels
- . Scenario 3 - A Class A, Level 1 system with 100 passages per day for identified vessels

Figure 4-10 presents, for each subfunction of each major procedure, the number of disc reads and writes required. Cases where either a read or write access is not required are indicated by a dash.

For the vessel and passage files, we have assumed a three level index structure. The fractional number of writes indicated in Figure 4-10 takes into consideration the fact that modifications to the index structure will be required (although infrequently for higher levels), when creating or deleting records in the file, as discussed in Section 4.2.

The total number of reads and writes required for each major function are also indicated. These numbers are then rounded to the next higher integer and used in subsequent figures for each scenario. Using USCG provided data for the number of executions per passage for each major function, we may calculate the number of reads and writes (and the sum of these) required for each scenario.

<u>ENTER VESSEL</u>	<u>READS</u>	<u>WRITES</u>
Search Index for Existing Vessel	3	-
Enter Vessel Name in Index	-	1.2
Check Free Block	1	-
Update Free Chain Pointers	-	1
Write New Record	-	1
Search Name Index and Insert Pointer	3	1
Insert in Call Sign Index	3	1.1
Insert in Lloyd's Index	3	1.1
TOTAL	<u>13</u>	<u>6.4</u>
 <u>DELETE VESSEL</u>		
Search Index for Vessel	3	-
Read Vessel Record	1	-
3 Search and Deletes @ 3, 1.1	9	3.3
Unlink from Active or Inactive Chain	2	2
Link to Free Chain	1	1
Update Free Chain Pointers	-	1
Write Blank Vessel Record	-	1
TOTAL	<u>16</u>	<u>8.3</u>
 <u>MODIFY VESSEL DATA</u>		
Search Index for Vessel	3	-
Update Vessel Data	<u>1</u>	<u>1</u>
TOTAL	4	1
 <u>ENTER NEW PASSAGE</u>		
Search Vessel Index	3	-
Read Vessel Data	1	-
Search Passage Index	3	-
Get Free Passage Block	3	3
Write New Passage Block	-	1
Insert in Name Index	3	1.1
Insert in ID Code Index	2	1.1
Insert in Pilot ID Index	2	1.1
Unlink Vessel from Inactive List	2	3
TOTAL	<u>19</u>	<u>10.3</u>
 <u>CHANGE STATUS</u>		
Search Passage Index	3	-
Read Passage Record	1	-
Write Updated Passage Record	-	1
TOTAL	<u>4</u>	<u>1</u>

Figure 4-10. Disc Access Estimates by Function
(Page 1 of 2)

<u>DELETE PASSAGE</u>	<u>READS</u>	<u>WRITES</u>
Search Index for Vessel	3	-
Read Record	1	-
Search and Delete from Indexes	7	3.3
Link Vessel Record to Inactive List	2	3
Link Passage Record to Free List	1	1
Write Blank Passage Record	-	1
TOTAL	14	8.3
 <u>IDENTIFY VESSEL</u>		
Search Passage Index	3	-
Read Passage Record	1	-
Delete Unidentified Record	3	3
Write New Passage Record	-	1
TOTAL	7	4
 <u>UPDATE VESSEL POSITION</u>		
Search Passage Index	3	-
Read and Write Passage Record	1	1
TOTAL	4	1
 <u>ENTER NEW COMMUNICATION</u>		
Search Passage Index	3	-
Read Passage Record	1	-
Read Previous Communications	2	-
Get Free Block (Blocking = 10)	-	.2
Write Pointer to New Block	-	.1
Write New Communication	-	1
TOTAL	6	1.3
 <u>MODIFY PASSAGE INFORMATION</u>		
Search Passage Index	3	-
Read and Write Passage Record	1	1
TOTAL	4	1

Figure 4-10. Disc Access Estimates by Function
(Page 2 of 2)

Figures 4-11 through 4-13 present the results of these calculations for scenarios 1 through 3 respectively. The total average number of disc accesses per day is:

<u>Scenario</u>	<u>Daily Disc Accesses</u>
1	287,520
2	27,670
3	19,460

FUNCTION	OPERATIONS PER DAY	PER FUNCTION		READS	WRITES	TOTAL	PER DAY		TOTAL
		READS	WRITES				WRITES	READS	
Enter New Vessel	180	13	7	20	2,340	1,260	3,600		
Modify Vessel Data	18	4	1	5	72	18	90		
Delete Vessel	180	16	9	25	2,880	1,620	4,500		
Enter New Passage	540	19	11	30	10,260	5,940	16,200		
Change Status	1,080	4	1	5	4,320	1,080	5,400		
Delete Passage	540	14	9	23	7,560	4,860	12,420		
Identify Vessel	810	7	4	11	5,670	3,240	8,910		
Update Vessel Position	3,600 ⁺	4	1	5	14,400	3,600	18,000		
Modify Passage Information	180	4	1	5	720	180	900		
Enter New Communication	4,500	6	2	8	27,000	9,000	36,000		
Scheduling/Routing	2,700	45	-	45	121,500	-	121,500		
Other Demand: Encounters, Local Traffic, etc.	6,000	10	-	10	60,000		60,000		
Total					256,722	30,798	287,520		
Peak Rate*					11.88/sec	1.43/sec			

*Four times average rate
+Four per vessel for Level 4 system

Figure 4-11. Disc Accesses - Watchstander Functions
Class C, Level 4 - 900 Vessels

	Operations Per Day	Per Function		Per Day		
		R	W	Reads	Writes	Total
Enter New Vessel	30	13	7	390	210	600
Modify Vessel Data	3	4	1	12	3	15
Delete Vessel	30	16	9	480	270	750
Enter New Passage	90	19	11	1710	990	2700
Change Status	180	4	1	720	180	900
Delete Passage	90	14	9	1260	810	2070
Identify Vessel	135	7	4	945	540	1485
Update Vessel Position	600 ⁺	4	1	2400	600	3000
Modify Passage Information	30	4	1	120	30	150
Enter New Communication	750	6	2	4500	1500	4500
Other Demand	1000	10	-	<u>10000</u>	<u>-</u>	<u>10000</u>
TOTAL				22537	5133	26170
Peak Rate*				1.04/sec	.24/sec	

*Four times average rate

+Four per identified vessel in Level 4 system

Figure 4-12. Disc Accesses - Watchstander Functions
Class B, Level 4 - 150 Identified Vessels

	Operations Per Day	Per Function		Reads	Writes	Total
		R	W			
Enter New Vessel	20	13	7	260	140	400
Modify Vessel Data	2	4	1	8	2	10
Delete Vessel	20	16	9	320	180	500
Enter New Passage	60	19	11	1140	660	1800
Change Status	120	4	1	480	120	600
Delete Passage	60	14	9	840	540	1380
Identify Vessel	-	7	4	-	-	-
Update Vessel Position	800	4	1	3200	800	4000
Modify Passage Information	20	4	1	80	20	100
Enter New Communication	500	6	2	3000	1000	4000
Other Demand	667	10	-	6670	-	6670
TOTAL				15998	3462	19460
Peak Rate*				.74/sec	.16/sec	

*Four times average rate.

Figure 4-13. Disc Accesses - Watchstander Functions
Class A, Level 1 - 100 Vessels

The Vessel Traffic Services Processing/Display Subsystem is a highly interactive system in which a substantial quantity of information must be rapidly communicated between the watchstanders and the computer system. Because of this highly interactive nature, the selection of display equipment and information formats is an extremely important consideration. For this reason the U. S. Coast Guard, outside the scope of this effort, intends to extensively study display technology and the human factors involved with human/computer interaction. This section provides a cursory look at system display requirements and display technology in order to provide a basis for making realistic assumptions about the probable nature of the displays and information formats.

5.1 VTS DISPLAY STATION REQUIREMENTS

5.1.1 Basic Features of VTS Display Subsystem

The basic requirements for the VTS watchstander's display station have been described in VTS Processing/Display Subsystem Functional Description. Each display station, at a minimum, will be composed of:

- 1) a communication group consisting of an alphanumeric keyboard, an alphanumeric display and a printer;
- 2) a traffic coordinator function console;
- 3) a vessel position display monitor;
- 4) an alert display.

Note: External communicator stations require only element 1 while traffic coordinator and watch supervisor stations require all four. The following discussions assume a traffic coordinator station unless otherwise indicated.

In the maximum possible configuration, the processing/display subsystem will support 15 separate stations as follows:

- . 1 watch supervisor station;
- . 10 traffic coordinator stations;
- . 3 external communicator stations.
- . 1 spare station which can be used for training

The watch supervisor's (WS) station and the watchstander training station are identical to the station of a traffic coordinator (TC). In event of a hardware failure, any TC station may be assigned as the WS station upon entry of the proper password.

The general characteristics required of each element are discussed in the VTS functional description. These characteristics may be satisfied by current display systems technology as discussed in Section 5.2. They are summarized in Table 1.

An important aspect of the VTS Display Station is the emphasis on man/machine communication. The workload for a traffic coordinator is a function of many variables. When the workload is heaviest, the objective is to make the operator's task as simple and easy as possible. One way is to prompt the operator when specific information or action is required. Another method, when multiple choices exist, is to present a list of such choices from which the operator selects the appropriate item. A third method is information feedback through which the operator is notified of status or completion of the requested action.

Beyond the basic hardware and software requirements are considerations dictated by good systems analysis and design practice. One aspect is the human engineering of the VTS display subsystem. The primary objective is to improve the watchstander's understanding of traffic flow and problems in the waterway. At the same time, the system acts as an extension of the watchstander by monitoring vessels in passage and alerting him to actual or potential emergencies or exceptions to established procedures. Another aspect is the dependability of the display hardware/software, which is related to system reliability, maintainability and availability. Also of major importance is the modularity of the system with its implications for flexibility in adding new functions and adapting to different environments (waterways) and traffic flows. Adaptability also implies expandability where new equipment may be

Alphanumeric Keyboard

- 'QWERTY' type
- numeric pad
- cursor positioning controls
- separate function pad

Alphanumeric Display

- 24 lines x 80 column screen
- 'touch' sensing
- scrolling
- multipage memory

Printer

- 80 columns
- upper/lower case

Vessel Position Display

- 1024 x 1024 point resolution
- x,y positioning
- 211 square inch usable screen surface

Traffic Coordinator Function Console

- function pad
- additional displays

Alert Display

- 22 lines by 40 columns
- alphanumeric display

Table 1. Basic Display Station
Hardware Requirements

integrated to handle increased workloads. Finally, in multi-sector waterways, the VTS system must coordinate the information distributed to watchstanders to ensure that each watchstander receives the appropriate current data for his sector with no information gap.

5.1.2 Display Station Function Categories

The elements of the display station will satisfy different operational functions. These functions can be categorized into four areas:

- . Data Entry

The data entry function is performed at all watchstander stations. The display unit only outputs data forms or operational instructions for the operator. The keyboard unit is used primarily to enter data into the VTS processing subsystem. Thus, good data editing and formatting functions are required. Examples of data input are ship names and positions, which are acquired by radio/telephone circuits, and weather data.

- . Data Retrieval

The data retrieval function is typified by large block transmissions of data from the VTS processing system to the operator. Usually, limited amounts of input data are used to identify desired data for retrieval. Display elements satisfying this function can be linked as slave units to master units used for interactive operations. An example of data retrieval is the accessing and formatting of a record in the vessel file for display to the operator. Another example is the preparation of reports summarizing traffic histories or shift operations.

. Inquiry-Response

The inquiry-response function is primarily an interactive short message communication between the operator and the VTS processing system. Extensive programming support is required for a broad, diverse and often complex spectrum of operations. Classes of operations include access to the data base, access to computation routines, access to training or 'help' routines and access to text manipulation features. The traffic coordinator will exercise all these operations from the master screen. Examples of inquiry-response are requests to display all ships in passage over 10,000 deadweight tons or requests to compute a dead-reckoned course.

. Monitoring

The monitoring function is typified by the vessel position monitor and alert displays. These display elements generally provide status information on critical elements that affect system operation. They are updated constantly in a real-time mode. In the VTS, the vessel position monitor continuously displays the position of each vessel in each sector of the waterway. The alert display will present information on detected hazard conditions. Because the displays are real-time, efficient communications support for data transmission in small packets is required. This support guarantees rapid response to the watchstander since his interactions with the vessel position monitor will usually involve only a few coordinates.

The extent to which the operations in the VTS fall into these categories will drive the generation, selection and specification of a display station configuration.

5.1.3 Human Engineering Considerations

The effectiveness of the VTS is highly dependent on the interaction between man and machine. Human engineering of the VTS display station will enhance the communications between man and machine, and aid the operators in evaluating and reacting to specific situations. In this section, three key elements of human engineering are discussed: response time, response types, and display station layout. Succeeding sections discuss aspects of data entry and data display including formatting and encoding problems.

Response Time

Response time is a critical factor in the engineering of a man-machine system. Response time is both a qualitative (psychological) factor and a quantitative (physical) factor. Psychologically, in communication, one usually expects a response - a feedback which continues the chain of thought. The user generally expects a response within a short period of time. Delay in response can interrupt the thought pattern and thus cause frustration and a feeling of loss of control. Physically, people organize work into segments which can be completed in a reasonable time interval. Interruptions in the activity or delays in completing the activity can be a frustrating experience.

Coupled with response time expectations is the nature of the problem solving environment. In a real-time control environment, such as VTS, complex problems are solved in single steps where the operator focuses attention on the immediate short term information quality and format. Distractions, "noise" or

abrupt shifts in thinking can affect the quality of subsequent decisions. Lengthy response times to operator commands will increase the probability of performance degrading attention shifts. As the response time increases, the mental efficiency drops. Lengthy response times also increase the base times necessary to perform a function.

Response time is a function of the operation to be performed by the operator request. Three categories of response times have been cited in the literature: update, request and display generation. Request response time is the time duration from completion of the request (e.g., pressing the function button) until the display appears. Update response time is the time between entry of new information into the system and the first appearance of that information in a display. Display generation response time is the time from a display request until the display can be completely viewed. It is associated with the preparation of complex displays such as formatted tables or maps.

Response time clearly affects the way in which operators can react to a situation and the way in which they will interact with the system. In general, it is not necessary to have the same response time for every operation. Different types of actions may be responded to with different speeds. Although the maximum response time for each action should be identified in the system design. Miller (1) has proposed a range of response times acceptable for system design. The following paragraphs represent a summary of his work with implications for system design.

- . Greater than 15 seconds - a response time greater than 15 seconds generally entices the operator to shift his attention to other activities. When he is ready, he should be able to conveniently access the displayed answer. A slave display upon which

the information is presented and remains until accessed, frees the operator from the task of constantly checking to see if his answer is ready.

- . 4 to 15 seconds - response times between 4 and 15 seconds are generally too long for an interactive dialogue between user and machine. A dedicated area of the display (as in split-screens) provides a convenient place to display the information when ready while allowing the operator to compose his next request.
- . 0 to 4 seconds - response times in this range are suitable for interactive dialogues. They allow the operator to maintain the mental set and concentration necessary to complete the task at hand. Interim responses such as a request acknowledgement prior to displaying the answer can maintain the continuity of attention and confidence in the operability of the system.
- . Almost Instantaneous - certain actions need an almost instantaneous response such as the pressing of a key and the appearance of a character on the screen. The motion of a light pen drawing a curve or deleting a line should be followed almost immediately by the appearance of a curve on the screen. Immediate responses to simple actions are reassurances of the reliability and utility of the system.

Nature of Responses

A complementary factor in human engineering is the nature of the response and its implication for operator reaction. Miller has discussed a series of response situations in the man/machine environment. A summary of these response types is presented in the following paragraphs:

- . Response to Initialization

Delays during system initialization are usually longer than the times indicated above. At a shift change, a new operator must sign-on and have his identity validated by password. Additionally, he may undergo a briefing by the system on the status of the sector for which he is responsible.

- . Error Messages

Error messages are displayed when a user mistake has been detected by the system. The user should be allowed to complete his current action (for example, typing a request) before he is interrupted by the error message. Lengthy displays should not be generated in a situation where they cannot be interrupted for an error message.

- . Response to Single Step Request

Responses to information requests depend on the complexity and amount of computation required. Good systems design requires that frequently issued requests, such as a request for the identity of a vessel at a checkpoint in a dead reckoning system, should be supported by efficient data structures. For

lengthy responses, partial data such as number of ships can be presented. An important factor is whether other work must stop until the response is received.

- . Keyboard Entry versus Light-Pen Entry

Operators generally expect faster responses from light pens than from entries generated by keyboard. The attention of the operator must be shifted from the keyboard to the screen whereas light pens are perceived to interact directly with the screen. To prevent irregular and lengthy responses to light pen entries, the graphics device is often isolated from the main system and supported by a dedicated processor.

- . Responses to Block Entries

For some actions, the user may enter a lengthy series of commands into the system and then direct that they be executed. Generally, these commands are interrelated, and thus, while interim responses may be expected and are acceptable, the operator is prepared to wait for the complete response. After a period of intense activity, the relaxation caused by a delay in response may be welcome.

Display Station Layout

The layout of the display station is a primary human engineering consideration. Terminal and graphics processor positioning should allow easy and natural use. The display station console should be designed to avoid unusual movements such as standing up to push a button. Displays requiring the watchstanders most immediate attention should be placed directly in front of him. Auxiliary displays providing supportive information should be positioned such that a simple turning motion of the watchstander's chair allows him to view the display directly. Keyboards and function consoles should be placed directly in front of the display screen where the output will appear. A possible configuration is shown in Figure 5-1.

5.1.4 Information Display Considerations

The configuration and number of components for the VTS display station depends, in part, on information display considerations. The objective is to present the watchstander with all the information necessary to analyze the situation without overloading him with information. Essential factors are the format of data in a display and cues presented to the operator prompting further action. This section discusses factors to be considered in developing information displays.

In an interactive environment, the information display format can substantially affect the efficiency of the watchstander. Badly formatted displays can lead to confusion and error on the part of the operator. Watchstander overload can be a particularly critical problem. In this case, the watchstander is presented with more messages and items than he can handle at one time. Some of this data may be superfluous to the decisions the operator is required to make and thus may obscure critical information necessary to the current situation. This is especially true if the system causes the watchstander to feel

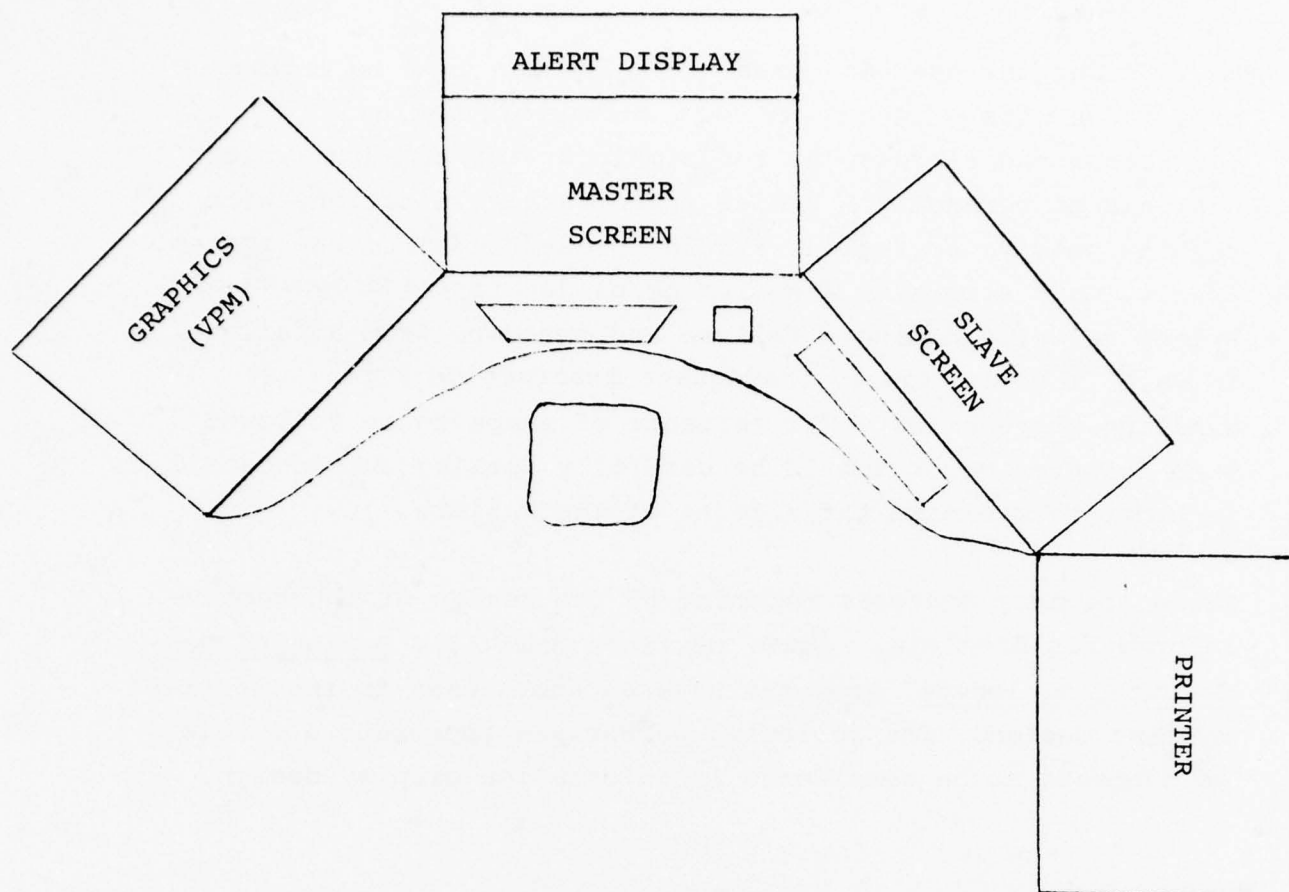


Figure 5-1

he is under intense time pressure which can lead to errors or poor judgements. Conversely, one must avoid making the watchstander's display too simple or too hum-drum which can lead to boredom. A proper mix of active functions with routine passive operations should be designed into the system. Lastly, many errors in a system occur due to the dropping of system security during a failure and recovery from a failure. In part, this is due to inadequate instruction for handling emergencies. The sequence of steps to be followed when failures occur should be carefully spelled out in detail in order to minimize the affects of the failure.

There are many concepts relating to the design of interactive information displays. James Martin's book (2), Design of Man-Computer Dialogues, presents several approaches to interactive systems design. The following paragraphs represent a synopsis of concepts to be considered in information display design.

- . Display a small amount of information at one time.
There is a tendency to fill a display screen with characters just because the space is available. If the information is rapidly changing, the operator may not be able to comprehend or respond to it in sufficient time. Thus, the amount of information on a screen should be minimized; highly summarized information assists in making rapid decisions. Additional data can be requested if it is needed to evaluate the situation.
- . Have one complete idea per display.
To minimize display clutter and suppress effects due to varying degrees of brightness, contrast blink rate, color discrimination, etc., only one major analysis should be contained in a display.

The alert display is a good example of this concept. Since the master screen can be augmented by data on a slave screen, the operator can be assured of adequate information.

- . The computer should always respond to the watchstander. Any input by the watchstander should initiate a response from the computer. Sometimes this input may be a simple acknowledgement of command acceptance (for example, cursor positioning). The operator should not be left hanging nor should he have to wait excessive amounts of time for the total display. Partial displays are reassuring and allow more time for evaluation.
- . Display clarity.
Lack of clarity can cause a display to be incorrectly interpreted. Careful attention to clear formatting is essential. An excellent approach is to use columnar displays with data justification depending on type. Masses of text should generally be avoided since the operator must roll his eyes to scan the text. Large textual displays also increase the clutter on the screen. Any message to the operator should be short and to the point. Proper labeling of all displays particularly column headings and graph axes is essential to information comprehension.
- . Display similarity.
The operator is usually confronted with a variety of displays. Displays should be similar in format, positioning, labeling criteria and color assignments to avoid attention shifts which would otherwise be required to understand a new display. Transitions between displays should be as smooth as possible particularly when complementary displays are presented on adjacent screens or in sequence.

- . Error correction.

The display terminal should provide an easy means for correcting errors. A backspace key can remove mistyped characters while closing up the text. A key should also be provided to cancel the current transaction.

- . Display encoding.

A variety of methods for encoding information on a display are available. They include flashing elements, varying brightness, underlining, color assignment and boxing of information. The key factor is to maintain a uniform set of criteria for encoding information across all displays. The number of criteria can be correlated with the number of capabilities provided by a particular terminal. Symbolology is a particularly important feature. Abbreviations should be minimized. The same name should represent the same information across all displays.

5.1.5 Display Element Selection

The previous sections have discussed some factors relevant to display systems technology. This section discusses the general factors for selection and specification of a display element.

- . Nature of Application

A display element should be suitable for the task it will satisfy. The essential tasks in an application should be identified and classified according to their nature.

- . System Objectives

The objectives of the system must be clearly defined. A range of different tasks may call for a variety of display elements to satisfy their requirements. Detailed technical design calculations may be necessary to

establish input/output rates, display element location and connection, and distribution to meet information response requirements. Among the objectives essential to VTS are:

- reduce the number of data errors entering system
- relieve clerical burdens
- provide up-to-date information displays
- provide fast turn-around of requests for commands
- increase operational efficiency

Communication with the system requires various inputs in order to retrieve information. Each display element should be selected to relieve the labor-intensity of the input task and thus reduce the possibility of errors. Data entry functions must be identified and classified. More common functions may be assisted by a display element which reduces the time-consuming input and validation of what is essentially static data. Variable data to be entered should be supported by display element features which reduce the time and effort for input. Alternative methods for inputting data should be available.

. Output

Information to be displayed should be identified and classified. Some data may be of transient interest while other items may require long term display or hard copy. Support for pre-designed forms can simplify the generation of displays. The output should be easy to read, sharply defined and clearly discernible. Flexibility in formatting the output should be provided to handle unusual situations or ad hoc requests.

. Transmission

The speed of transmission must be sufficient for both input and output. Buffering of data by the display element can make transmission more efficient. The display element should be compatible with the computer and not require excessive encoding. The speed should not place an undue burden on the processor or on the operator.

. Intelligence

Providing intelligence in a display element is usually more expensive than placing it in the host processor. The location of intelligence in a system is a function of cost, complexity and task requirements. However, the use of intelligent display elements can provide such features as:

- data editing and checking
- reduced transmission costs
- error control
- security
- decreased utilization of other processors
- more flexibility in usage and expansion

. Security

The prevention of unauthorized access to or use of a display element can be an important criterion. Physical security might include a lock and a key which is required to activate the element. Authentication by password or code may be provided by intelligent display elements. Access to selected files, processes or other components of a system can be associated with a specific display element.

. Compatibility

An essential criterion is the compatibility between display element and computer. The cost of designing a custom interface or special software may far exceed the utility gained from a particular component. Intelligent display elements can obviate some difficulties through emulation or reprogramming. While there may be good reasons for installing 'mixed' hardware, it is necessary to ascertain without delay the cause of any malfunction and where the responsibility lies for correcting the fault.

. Expandability

Flexibility to adapt to change should be built into the system. The system should be designed so that it can be expanded to handle increased volume of throughput, additional applications or modified procedures or take advantage of new technology. The tradeoff between rewriting software and replacing display elements or delaying system enhancements must be carefully evaluated.

. Reliability

Most display elements are fairly reliable. Components are becoming smaller with fewer mechanical parts. Even so, faults do occur, and fallback procedures must be available. The cost and availability of maintenance services must be considered for each location of a VTS system.

. Cost

Each of the above factors may be reflected in the cost of the display elements. It may be better to pay a little extra for a component that is better suited to a particular task than to 'save' money by acquiring a cheaper, less flexible model. It should be clear, however, that increased cost does not necessarily improve the utility of a display element for a particular application such as VTS.

5.2 SURVEY OF DISPLAY TECHNOLOGY

This section surveys the features and characteristics of current display technology. A substantial number of commercial vendors supply display hardware of various levels of sophistication that can be interfaced with many micro- and mini-computer systems. Datapro⁽³⁾ provides comprehensive evaluations of many of the vendors.

5.2.1 Structure of the Display Station

The VTS display station will require a variety of components for implementing the various types of displays. The Vessel Position Monitor will require a graphics processor with at least a 1024 x 1024 point display. Associated with the VPM will be a manipulation device consisting of either a light/sonic pen, a trackball or a joy stick. Color is a desirable feature but cost and other considerations may preclude its use. If color is selected, a maximum of four colors will be used to prevent chromatic overload on the operator.

For information displays, a minimum of one alphanumeric terminal and an alphanumeric keyboard is required. This terminal is the master communications display for the operator. The terminal must display a minimum of 24 lines by 80 characters of information. During normal operations, it is expected that information to and from the master screen will generally consist of short messages. It is possible that the operator may wish to display the entire vessel information record as a reference. To do so would require a replacement of the current master screen contents. It may therefore be desirable to consider a slave screen to be used for lengthy displays. The slave screen would be changed only by a specific request of the operator whereas the master screens contents change dynamically as new commands are executed and situations occur. Thus, a display can be generated on a slave screen as a reference table until no longer needed.

The alert display terminal provides, at a minimum, a display of 20 lines x 40 characters plus header information. Alert displays are automatically generated by the system and are not under direct operator control. For simplicity, it is assumed that the alert display is similar to the master/slave screen characteristics.

A printer may be required to produce hard copy reproductions of master/slave screen contents. The printer must print a minimum of 80 columns in both upper and lower case. An obvious engineering consideration is that the printer be as quiet as possible since it will be colocated with the display station.

Finally, the traffic coordinator function console (TCFC) requires a keyboard which allows the watchstander to execute frequently used functions at the push of a button. The TCFC will probably not be available directly off-the-shelf but fabricated from off-the-shelf components.

5.2.2 Printer Technology

Printers which may be applicable to the VTS Processing/Display Subsystem display stations consist of "receive only" printers (ROP) or are combined with a keyboard to form a printer terminal. A ROP is often used to provide a selective hard copy display in conjunction with a video display unit (VDU). Printers can be characterized as either character at a time or line at a time printers. The speed of character printers varies between 10 and 120 characters per second. Line printers generally range in speed from 150 to greater than 1100 lines per minute. The higher the output speed, the greater the cost of the device. Line printers tend to be much noisier than character printers particularly when they use impact technology.

A teleprinter may be of the impact or non-impact type. Impact printers press a typeface or a matrix of dots against paper through an intervening inked ribbon. The force of impact is strong enough to generate multiple copies although it is rather noisy. Non-impact printers generally use the inkjet, electrothermal or xerographic techniques to produce an image upon the paper. These printers do not provide a capability for carbon copies and the quality of the printed image is not normally as good as impact printers.

Impact teleprinters frequently employ a serial one character at a time printing technique. Full character typefaces produce highly legible images in a number of appealing fonts. But, they do not lend themselves to printing speeds above 30 characters per second because of the complex mechanical arrangement necessary to select a character, position the print mechanism, and strike the printed image. An alternate method is the matrix printer which represents a compromise between decreased character legibility and substantially higher print speeds (in excess of 100 characters per second). The matrix method formulates a printed image from a rectangular matrix of dots. Matrix printers are subject to a greater amount of wear within the print head as a result of the succession of pin movements required to create each character.

Non-impact printers use three major methods of image production. The electrothermal printer technique generates heat in a typeface element which interacts with heat sensitive paper to form the character. The ink-jet technique sprays a stream of electronically charged ink droplets onto ordinary paper to produce printed characters. Character formation is controlled by electrostatic deflection plates. This technique is relatively

expensive. The last technique, also expensive, is the xerographic method. Reliability of most non-impact printers is comparatively high because they have few mechanical parts. The most desirable feature of non-impact printers is that they are quiet -- an important consideration in the VTS environment.

The paper used by the printer deserves consideration as a design element. The paper must be suitable for subsequent use - thus, it must be determined whether to use continuous roll or perforated fanfold paper. The use of perforated fanfold paper calls for a 'sprocket feed' mechanism to ensure correct alignment; in order cases a 'friction feed' mechanism may be adequate. Another design element is the horizontal spacing of characters (usually 10 to 12 per inch) and vertical spacing (6 and 8 lines per inch are common). These spaces can affect the visual perception and acuity of the output.

As mentioned above, impact printers and high speed line printers are often noisy. The trade-off between speed and noise level in the VTS environment deserves careful attention, since it is expected that teleprinters will be colocated with the display stations.

A recent development in teleprinter technology is the inclusion of microprocessors as control units. The control programs support basic functions of the teleprinter and provide the ability to emulate the telecommunications protocol employed by other terminals.

The typical characteristics of impact versus non-impact printer technology are summarized in the table below.

PRINT CHARACTERISTIC	IMPACT	NON-IMPACT
Operation	Noisy	Quiet
Speed	Usually Slow	Potentially Fast
Print Quality	High to Medium	Medium to Low
Printed Copies	Multiple	Single
Paper Type	Ordinary	Typically Special
Reliability	Low to Medium	Potentially High
Cost	High to Medium	Medium to Low
Physical Size	Medium to Large	Small

In selecting a teleprinter terminal for the VTS display station, the following checklist of factors should be considered.

- . Are both upper/lower case letters required/provided?
- . Is the terminal easy to use?
- . Is impact or non-impact printing more suitable?
- . Are multiple copies of printout required?
- . Is the printer fast enough?
- . Are the printed characters easy to read?
- . Is printing in two colors desirable?

- . How many characters per line must be printed?
- . Is the terminal noisy?
- . What type of paper is required/provided?
- . Does the paper required present a costly expense?
- . What transmission code is used?
- . Is a buffered terminal needed?
- . Are the terminal and the computer compatible?
(i.e., does there exist an interface?)
- . Should the terminal be portable?
- . Is an end-of-line indicator required/provided?
- . Are tabulation features required?

5.2.3 Video Display Technology

A video display unit (VDU) consists basically of a keyboard for the manual input of characters, and a cathode ray tube (CRT) screen which displays the characters held in the terminal's character storage. Alphanumeric VDU's have replaced teleprinters as the primary man-machine interface at most levels of system interaction. This fact, coupled with other characteristics of VDUs, has provided the impetus for a staggering array of VDU terminals in the marketplace.

A VDU terminal can be assigned to one of three general classes (ignoring specialized/customized terminals):

- . dumb terminals which offer a limited number of functions; most feature teletype compatibility;
- . quasi-intelligent terminals which offer functions such as editing and formatted data entry
- . programmable terminals which feature extended software support in varying degrees of sophistication.

The cost of a video display unit is proportional to its capability. Dumb terminals typically range between \$1,000 and \$2,000 while programmable terminals range upward from \$6,000. Most programmable terminals can accommodate an array of peripherals which increase their cost significantly.

Most VDU terminals have a keyboard for manual entry of data. VDUs, unlike teleprinters, normally have a buffer or memory store for supporting the refreshing of the screen image. Storage tube displays store the information directly on the screen in analog form, thus requiring no buffer and having no flicker. VDUs with buffers transmit data in three possible ways. In half-duplex operation, each character is displayed and transmitted as it is keyed in. In full-duplex operation, each character is transmitted as it is keyed in and reflected back to the terminal from the computer. Finally, in block-mode operation, the entire contents of the buffer is transmitted to the computer.

Most of the display terminals introduced during the past two years have been controlled by a built-in microprocessor. The control program usually resides in read only memory (ROM) or programmable read only memory (PROM). ROM - resident programs are permanent, while PROM programs are erasable, although this is an involved process. The programs control the terminals basic functions as well as emulating various telecommunications protocols. Microprocessors also allow the meaning of the function keys to be easily changed by replacing the PROM. In programmable terminals, users can add new capabilities to the terminal by loading a program into its memory. These terminals also support a variety of peripherals such as cassettes and diskettes. The power of the programmable terminals may prove cost-effective in certain distributed environments.

VDU terminals are characterized by a number of features. One such feature is the ability to display images in color to identify certain conditions or types of data. Up to eight colors are available on some models with a significant increase in cost. The reverse video feature displays a negative image of the data, i.e., black on white instead of white on black. This feature is useful for setting off fields of interest in a display. Another feature is programmable brightness levels which allows visual separation of information by varying the intensity level of the display. Finally, a character or field may be set to blink in order to attract attention. Critical entries on the alert display could blink to indicate the urgency of the situation.

For VDUs which can operate in a block mode, two additional features are often provided. The scroll feature moves all displayed lines of data up or down by one line as the new line is added and an existing one removed. Typically, data is lost as is rolled off the screen but a buffer could allow these lines to be saved. If the buffer can store multiple screens of data, then paging is often provided - that is, a user can see any of the stored pages for display.

Usually, a screen is blank when the system is initialized. To indicate the current position on the screen, a character called a 'cursor' is displayed where the next character is to be keyed in or where a character will be deleted or removed. The cursor is generally a horizontal bar located below the normal character position so as not to obscure any characters. Sometimes the cursor can be set to blink to attract attention or be suppressed altogether. The cursor is usually non-destructive in that it does not delete existing characters over which it passes.

Usually associated with the cursor are editing and formatting controls. Special keys are usually provided to position the cursor in the following manner:

- . move the cursor left/right one space;
- . move the cursor up/down one line;
- . move the cursor to first position, top line;
- . move the cursor to first position, last line;

- . tab the cursor forward/backward to a previously set tab stop;
- . move the cursor to the first position of the next line;
- . backspace the cursor one space to the left (may automatically delete any character);
- . move the cursor to same position, next line.

The editing features, if present, consist of a set of functions for manipulating elements of a screen image. Among the more common functions performed relative to the cursor are:

- . INSERT-permits characters to be added to an existing line by shifting the characters already displayed to the right.
- . INSERT LINE-allows a new line to be added above the current position of the cursor.
- . DELETE-removes characters from an existing line with the remaining text closing up the gap after the characters are deleted.
- . LINE DELETE-allows lines of text to be removed.
- . CLEAR DISPLAY-erases all displayed data on the screen and returns the cursor to the home position.
- . SET BLINK-initiates blinking of data inserted at the position occupied by the cursor.
- . CLEAR BLINK-terminates blinking at the position occupied by the cursor.

On some models of VDUs it is possible to display fixed 'forms' on the screen. The user enters data into the fields of the form with the VDU automatically moving the cursor to the next empty field after each entry. Some terminals offer a protected and edited format capability for this 'fill-in-the-blanks' approach. Users may not enter data into positions occupied by elements of the form.

The presentation of the characters displayed on the screen is important. Unusual screen angles should be avoided in order to prevent eye strain due to glare or awkward viewing patterns. Characters on the screen should be easily distinguishable. Screens vary in size and the number of lines and columns they can display. Standard screens usually display 80 characters per line with 12 to 25 lines available. This yields a range of 960 to 2000 characters of data on the screen.

Character generation is accomplished either by the dot matrix or a vector generator. The dot matrix method illuminates selected dots from an array (typically 5 x 7) to form the character. The vector generator moves the beam from one coordinate position to the next, with each increment treated as a quantum step in the X or Y axis or both. All characters are then composed of straight line segments. Figure 5-2 depicts possible character formations.

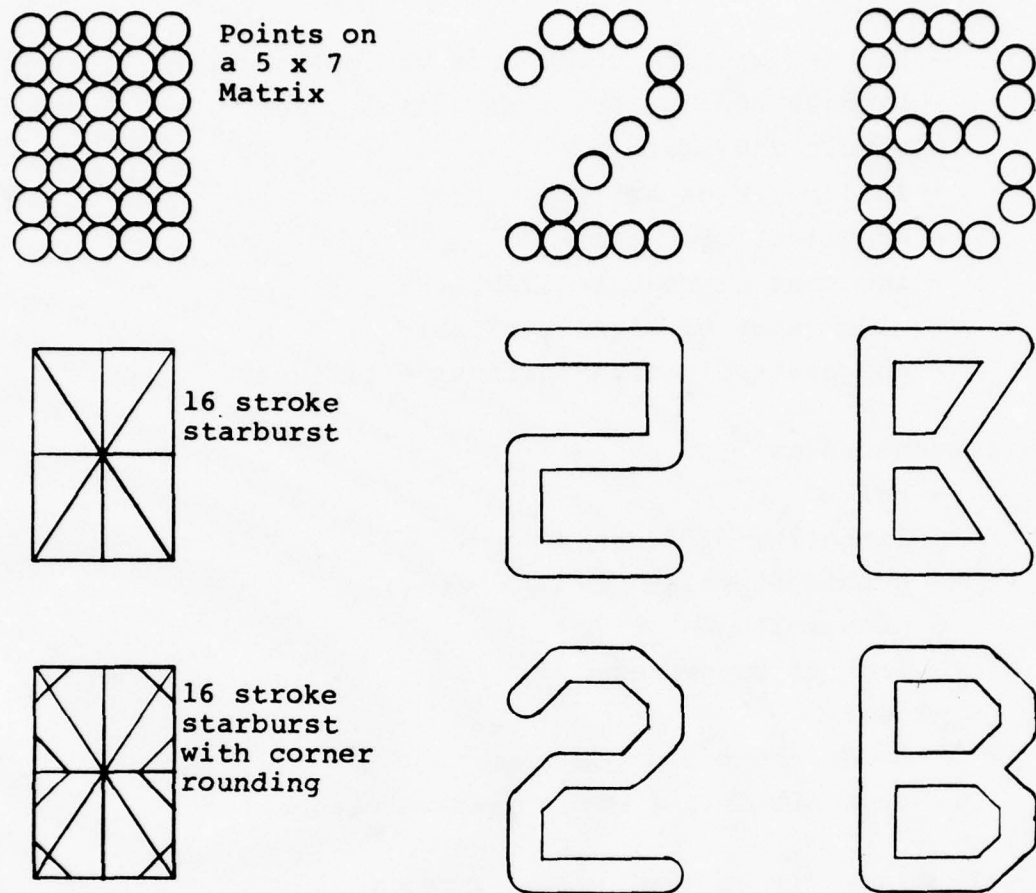


Figure 5-2. Character Generation by Dot Matrix
and Vector Methods

Finally, the display brightness should not dazzle the viewer or obscure the data by blurring. As brightness is increased, the flicker becomes more noticable and thus requires a higher refresh rate. A flicker-free image is one that appears steady and stable to the viewer and thus provides a sense of solidity to the display. Flickering displays can prove to be extremely distracting and tiring to the operator.

The advantages and disadvantages of VDUs are presented in the following table:

. Advantages

- highest writing speed of any display device
- highest resolution of any display device
- simple addressing
- full color capability
- excellent gray scale
- inherent storage available
- wide range of sizes available
- comparatively high luminous efficiency

. Disadvantages

- bulk
- linearity of distortion
- curved faceplate in some areas
- high voltages
- lack of ruggedness
- cost
- power (heat dissipation)
- approximately 1 meter maximum diagonal size

When selecting a video display unit, the following checklist of selection factors can provide a guideline:

- . Are the available editing facilities suitable and easy-to-use?
- . Are there special keys available/required?
- . Is the keyboard repertoire suitable?
- . Are both upper/lower case letters required/provided?
- . If lower case is provided, is it of sufficient quality?
- . Is the keyboard physically compatible with the VDU?

- . How many lines per display and characters per line are required/provided?
- . Is page or roll mode for output presentation required/provided?
- . Is the display easy to read with characters clearly separated?
- . What other peripherals are required/provided/supported?
- . Are protected fields required/provided?
- . Are the cursor controls suitable?
- . Can cursor position be sensed or controlled?
- . Is the cursor easily distinguishable on the screen?
- . Is the display screen flicker-free?
- . Is display brightness variable to suit the operator?
- . Are selectable horizontal tabs required/provided?
- . Is color required/provided? Other options?
- . Are the features of the terminal changeable, i.e., microprogrammed?
- . Is the buffer of the terminal suitable?
- . If programmable, where do the final programs reside and how are they loaded/reloaded?

5.2.4 Graphics Display Technology

Graphics displays are similar to video display units in appearance and function. The display tube does not constitute the primary difference; rather, it is the controlling logic. The image on a graphics display is an image constructed of lines connecting addresses in a matrix or matrix points illuminated by a scanning beam. These lines are the results of logical instructions residing in the memory of the graphics processor. The graphic image can be manipulated and changed by the graphics processor thus providing a significant increase in flexibility over VDUs and a powerful, versatile man/machine interface.

A key concept in graphics displays is the fact that the positions of the elements comprising the information representation are themselves important information elements. In the graph of vessel movement, the position carries much of the information. Different vessels may be identified by different symbols thus providing data set identification. The main point regarding graphics displays is whether the required graphic information can be clearly and economically displayed and manipulated.

A user's association with a graphics display can be either:

- . passive - one just views the graphics presentations;
- . active - one participates in the generation or modification of the presentations.

The degree of interaction and complexity of presentations determines the sophistication of the device and supporting computing facilities. The sophistication of interaction is controlled by the programming of the graphics processor. Whatever the functional extent, the user must identify what he wants done, where it should be done, and supply any data necessary to accomplish the task.

Graphics devices come in a variety of types. Prices range from \$4,000 - \$15,000 at the low end for a 'dumb' graphics terminal up to \$40,000 - \$100,000 for sophisticated microprocessor-controlled systems with extensive capabilities. The features of graphics devices are discussed in the following paragraphs.

One feature available for graphics displays is color. Color adds an important dimension to graphics displays in that it provides additional information. This extra dimension is contrast which is evident in two related ways. The first is more rapid understanding of information presented to the operator. For example, vessels placed in alert status could be colored red to signify to the operator a critical situation. The second aspect is the ability of the operator to distinguish details of the display. Coloring alert status vessels red would allow the operator to identify all such vessels at a glance. Graphics displays on the market can support a variety of colors from the basic black and white up to eight colors with varying combinations.

Another feature is the viewing area or size of the display. Normally, the addressable matrix is 512 x 512 distinct points. Some systems provide up to 4096 x 4096 point matrices. The screen size and the size of the viewable matrix is a function of the resolution required for the display. Larger screen sizes may require proportionately more software support from the graphics processor.

Another feature is the capability to 'zoom' the display, which is the ability to automatically change the scaling factor for the displayed image. The effect is to enlarge a portion of the display to provide additional details. The zoom function may be implemented in hardware or software although the latter gives more flexibility.

Graphics displays use either raster scan or random position to generate images. Random positioning offers high resolution but is limited in color capability and supports a relatively flicker-free information content. In this technique, the CRT beam can be moved in any direction directly to the picture element to be displayed. In raster scanning, the beam moves horizontally across alternate lines of the entire frame during one downward scan then returns to the top in a continuous regular pattern. Raster scanning provides high color presentation and high flicker-free content at the expense of resolution. TV, the most common raster system, employs a 512 x 512 matrix as opposed to 1024 x 1024 matrix used with most random positioning systems.

Graphics display processors can use either the point plot or vector generation method for producing a display. In the point-plot method, an x, y coordinate is identified along with a z-axis intensity. In vector generation, the x,y coordinates for the endpoints of a line segment are specified; the beam traces out the line segment. Vector generation may be either stroke or incremental where the latter moves in distinct quantum steps of x,y coordinates. In many systems, character generators are provided to relieve the processor of the software burden. Usually, a 64-character set includes upper case alphabets, numerics and a few special symbols.

The graphics processor may be either an external central processing unit such as a minicomputer or an integral part of the graphics system. Integrated graphics systems provide a workload sharing capability as well as a standardized interface to the graphics functions. Most vendors of graphics systems provide software packages with their systems.

Among the software support functions to be considered in a graphics display system are:

- . LINE - to draw a line segment between two points
- . ROTATE - to rotate a point (x,y) through a clockwise angle θ , about a selected origin.
- . SCALE - to expand/contract the size of the image
- . CLIPPING/WINDOWING - to extract a subpicture from a total image
- . BOXING - computing the size and location of a window image
- . MOVE - move the beam a specified distance
- . POINT - display point at X,Y

Finally, careful consideration should be given to the method for entering data into the graphics processor. Most graphics displays include the standard QWERTY keyboard with an additional numeric pad. In addition, special function keys may be provided for certain graphics functions such as zoom. To manipulate the cursor in a graphics systems, and thus support operator interaction, three basic data entry devices are available:

- . The joy stick is a device that looks like a large toggle switch. Moving the stick left/right and up/down moves the cursor in two dimensions on the display. Once the cursor has been positioned at a point, depression of a key on the joy stick will cause the coordinates to be transferred to the computer.

- . a tracking ball works in essentially the same manner although the key is located on a separate keypad. The tracking ball allows a full 360° spherical tracking capability. The difference in the two devices is in the implementation: the joystick uses a gear mechanism while the track ball uses phase encoders.
- . A "mouse" can also be used for cursor positioning. It is a hand held device which is moved over a surface. The X and Y coordinates of the motion are detected by two wheels on the bottom which are oriented at right angles to each other.
- . A light/sonic pen allows the operator to identify a point on the screen which is then illuminated by the computer. Tracking must be accomplished by a software module.

When selecting a graphics display system, many of the factors pertinent to VDUs apply also to the graphics display. In this checklist, we mention only those factors directly applicable to graphics systems.

- . Where will the control software reside - internal/external to the processor?
- . Can controllers be shared?
- . What functions are required of the graphics software?
- . What mode of image generation is required - is it related to screen size?

5.2.5 State-Of-The-Art Technology

In the past two decades, display technology has progressed to its current level of sophistication at a rate that rivals the explosive growth of digital computer systems. The state-of-the-art provides a variety of display types for effective communication of information. This section discusses other elements which should be considered for the VTS display station.

Three basic, functional categories of digital displays exist:

- . Numeric readouts which are designed primarily to portray numerical values. They are used singly or in assemblies. They include LEDs, liquid crystals, incandescents and electroluminescents. They normally utilize 7 bar segments to form characters although 5 x 7 dot matrices or fully formed, filamentary figures (NIXIE TUBES) have been used.
- . Alphanumeric readouts which are more complex than numeric readouts since they display an entire character set, 10 numerals and several punctuation marks. The minimum format required is a 5 x 7 dot matrix or 13 bar segments. Attempting to display large amounts of text is quite difficult since the number of display elements and associated electronics increases proportionately with the size of the display.
- . Multiple character/line generator panels which generally display from 16 to 256 characters of randomly generated information. They are quite versatile but their viewing distance is limited to 20 feet or less. They require more sophisticated electronics but can be multiplexed to reduce overall power consumption.

Selection of display technology at the state-of-the-art is a function of cost. Cost is composed of a number of factors including:

- . cost of display
- . driver/decoder electronics
- . related printed circuit board design, fabrication, assembly and test
- . interconnection
- . polarizers, bezels or other contrast enhancement filters
- . special power requirements
- . possible radio frequency or electromagnetic interference
- . display driver software

A number of tradeoffs bear on the display cost and are related to the environment in which the display is to be used. Some of these tradeoffs are:

- . character size versus viewing distance
- . ambient light versus brightness and contrast
- . power consumption
- . simplicity of electronics
- . panel efficiency and panel depth
- . facility of comprehension and use of color
- . human factors
- . reliability and maintainability

There are several major technologies available for displays. The following paragraphs summarize these technologies.

- . LED (light emitting diode) is the most popular class. They are general IC - compatible, compact, easily multiplexed, cheap and simply assembled in multi-digit arrays. LEDs generally average about 120 mW of power. Because they are solid-state, they are extremely reliable and resistant to shock and

vibration. Their primary disadvantages are a requirement for a constant current source and susceptibility to thermal failure if overheated. LEDs are limited in luminous intensity which precludes their use in high ambient lighting and long viewing distances. Maximum viewing distance is 3 - 60 feet with maximum viewing angle 90-160 degrees.

- . Incandescents are the mainstay of display technology. They have several advantages including TTL compatibility, relatively long life, controllable brightness and easy filtering to provide all colors. They are capable of operating over broad temperature ranges and can output up to 13,000 feet lamberts making them easy to read in direct sunlight. Incandescents are difficult to multiplex and consume up to 1 watt of power per digit. Incandescents with filamentary segments or glass envelopes are susceptible to damage from shock and vibration. Viewing distance can range up to 150 feet.
- . Liquid crystal displays are a recently emergent innovation which provide the lowest-powered units (less than 500 microwatts) and possibly lowest cost per digit. They usually come in small, compact, slim, multiple-digit assemblies and are easily read in high ambient lighting conditions. Reliability and life expectancy are currently problems as is fragility due to their glass construction. Interconnection is difficult and they are not easily multiplexed. While they are MOS compatible, they generally require more electronics to drive multidigit arrays. Viewing distance is about 25 feet maximum.

. Gas discharge readouts are extensively used in small multiple instrumentation. Their advantages include high brightness, uniform color and character, and ease of multiplexing. Easily readable in high ambient lighting, they have life expectancies up to 10 years. However, voltages in the range of 200 to 300 volts are required for their operation. Their glass construction makes them susceptible to shock and vibration. One variant is a message panel gas discharge device which can accommodate from 16 to 600 characters of data utilizing dot matrix formats. These message panels provide simplicity of addressing, small size and volume and are quite versatile. Another variant is the plasma panel which replaces one glass dielectric wall of a message panel with a photo conductive, glass composite layer. This allows the gas in individual cells to be triggered by an external light source. This yields a near-indefinite stay-on time and provides for selective erasure by light or pulsed operation. Plasma panels provide slim, compact message displays in the range of 256 to 4000 characters.

5.3 DESIGN ASSUMPTIONS

From the preceding discussion of display requirements and technology, it can easily be seen that there are a substantial number of devices that could be used and a broad range of technology which can be applied to the design of a display station for use in the VTS Processing/Display Subsystem. Selection of the devices to be used and the design of the display station involves careful consideration of human factors as well as numerous engineering considerations.

The design of the display stations will play a significant part in determining the overall effectiveness of the VTS System. Because of the potentially large multipliers involved, the design of the display stations may also contribute a substantial percentage of the total VTS System cost.

Because of the importance of the design of the display station, we will attempt to make basic assumptions which will allow us to properly evaluate alternative architectures without unnecessarily limiting the options available for the design of the display station. In this section we will discuss these basic assumptions.

5.3.1 Display Station Processor

Since distributed processing is a fundamental concept in the design of the VTS Processing/Display Subsystem it is appropriate to assume that a processor will be used to support the various devices associated with the display station and provide an interface with the balance of the system.

The display station processor will handle the processing necessary for data entry and display and report formatting. It will manage the I/O associated with standard and special function keyboards, the printer, CRTs, and the graphics unit. Display station processors will interface with the central part of the system by a shared bus, direct serial lines, or by some other device depending on the selected architecture.

The assumption of a display station processor simplifies the architecture evaluation while providing the highest possible degree of flexibility in the actual design of the display station. The flexibility remains to use one graphics unit or two, one CRT or several, a printer slaved to a CRT, an independent printer, or no printer at all. Graphics units can be "dumb" requiring significant support from the display station processor or highly intelligent requiring little support.

The precise characteristics of these processors cannot be determined until the actual devices to be supported have been selected. Perhaps the most significant issue to be resolved is the amount of support required for the graphics unit or units since graphics support can require substantial processing and memory.

Since the design of the display stations and their processors will be independent of the architecture selected (except for the interconnection with other system processors) precise sizing of these processors is unnecessary at this point in the design. This allows the evaluation of alternative architectures to continue without making design choices that would impact the ability to properly design the display station in the light of human factors studies which will be considered apart from this current effort.

5.3.2 Communications Loads

While the power and memory for the display station processor need not be specified at this time, it is necessary to determine the communications required between the display station processors and other processors in the system.

The following is a list of the major communications loads which must be considered in the design of systems for the alternative architecture.

Vessel Position Data

Vessel position data will be generated by radar, other sensors and/or manual inputs. These updates of vessel positions will be transmitted to the appropriate display stations. The number of vessels to be displayed and hence updated periodically will be a function of the total number of vessels handled and the number of stations. Figures for this load will be included in the evaluation of the architectures based on the assumed characteristics of the various classes and levels of systems.

Watchstander Requests

Data input by the watchstander and requests by watchstanders for data and reports will account for a portion of the total communications load. Because these loads result from human action, the rate at which they can be generated is low. A peak rate for these inputs is expected to be 25 bytes per second.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 30,000 bytes of data must be transmitted in a six second time period to replace a display. This assumes approximately three bytes for each of 10,000 vectors.

REFERENCE

1. Robert B. Miller, "Response Time in Man-Computer Conversational Transactions", AFIPS Conference Proceedings (Fall Joint Computer Conference 1968), Thompson Book Company, Washington, D. C., 1968.
2. James Martin, Design of Man-Computer Dialogues, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.
3. Datapro 70, "All About Graphic Display Devices", Datapro Research Corporation, Delran, New Jersey, March, 1977.

6.1 PERSPECTIVE

For about 20 years various forms of multiple processor systems have been constructed in an attempt to achieve one or more of the following goals:

- . Increased throughput
- . Increased system integrity and availability
- . Sharing of resources associated with the individual processors
- . Modular expandability
- . Improved flexibility
- . Reduced system costs

Minicomputers and microprocessors have significantly increased interest in multiple processor systems as researchers and system designers seek to create improved computer systems.

A variety of multiple processor systems have been studied by various investigators. Numerous claims have been made about the efficacy of multiple processor systems. Indeed if all the claims which have been made were accepted one could begin to believe that it is possible to build the "ideal" computer system which would be completely flexible and cost effective for almost any application. We hasten to point out, however, that such a system has not been

built and there is little reason to expect that this "ideal" system will be forthcoming in the foreseeable future.

The "ideal" system cannot currently be constructed since in the general case the approaches which have been attempted to date have had weaknesses to counterbalance their strengths. It is possible, however, to achieve a substantial portion of the potential benefits of multiple computer systems. These benefits can be realized by matching system architectures to particular applications. By matching architectures and applications we can select approaches that take advantage of an architecture's strengths thus reaping many of the potential benefits. At the same time we can select an architecture with weaknesses which may be significant in many applications, but are of little consequence to the application at hand.

Our effort then will be directed toward matching the VTS application to one of the many system architectures which are available for consideration. In the remainder of this section we will survey the system architectures which may be appropriate for use in the VTS Processing/Display Subsystem. In subsequent sections, the four most promising architectures will be selected. Additional analysis will be performed to assist the U. S. Coast Guard in selecting one of the four system architectures for use in the VTS.

6.2 WHAT IS A SYSTEM ARCHITECTURE?

Before we discuss alternative architectures it is appropriate to define what we mean by system architecture. Simply stated a system architecture, as we are using the term here, is the combination of the logical and physical structure of the system.

In the review which follows we will concentrate on three major aspects of the system architecture which include the type of processors, the type and structure of the interconnection mechanism, and the way in which the functions of the system are assigned to the processors. Each of these factors will be explored in some detail in the sections which follow.

6.3 PROCESSORS

Processors today cover a broad spectrum from single chip microprocessors to the array processors of the "super computers." Processors which could be considered for VTS are summarized below.

6.3.1 Eight Bit Microprocessors

Eight bit microprocessors are in use in a variety of systems and components today. Microprocessors are available as single chip processors and as a part of single board computers. The cost of the single chip processors has steadily declined creating a revolution of sorts in digital design as microprocessors continue to replace hardwired logic in a vast number of applications. The microprocessor has also opened up a whole new area of computing: the hobbyist computer.

Eight bit microprocessors have a number of limitations, however. Execution is slow. The instruction sets are limited and supporting software is limited or non-existent. Substantial hardware development is also frequently required in using microprocessors although some small, but relatively complete systems are available. Peripherals for such systems are often not available or require special hardware to interface.

6.3.2 Sixteen Bit Micros and Small Minis

Sixteen bit micros and small minis form a second class of processors which are currently being marketed primarily by the minicomputer manufacturers as the low end of the minis. Most are upward compatible with their more powerful minicomputer cousins.

Execution is faster on the 16 bit than on the eight bit micro-processors. Using the instruction basis on which the processing model in Section 3 was developed, a typical 16 bit micro can handle the equivalent of approximately 600,000 instruction cycles per second which is roughly half the processing power of the standard mini.

Many of the manufacturers do not currently offer memory expansion beyond 64K bytes, although a few offer up to 128K bytes of memory. Floating point hardware is also not available for most of the low end systems.

Software support is generally good and a wide range of peripherals are available.

6.3.3 Standard Minicomputers

Minicomputers are available from a variety of manufacturers and have been used in a wide range of applications. These processors can handle approximately 1.25 million instruction cycles per second in the normal configuration and a number of options such as cache memory and memory interleaving are available to increase processing power.

Memory expansion to 500K bytes or more is generally available and essentially any peripheral can be interfaced to today's standard minis.

Both floating point hardware and firmware are common and extensive software is available including higher order languages, operating systems, and data base management systems.

6.3.4 Thirty-Two Bit Minicomputers

Several manufacturers are now marketing 32 bit minicomputers to compete both with other minis and with a number of the large mainframes.

These systems feature expanded instruction sets and significantly greater address space. One million bytes of memory or more can be supported without memory mapping which is normally used with the 16 bit minis.

Peripheral and software availability is comparable to standard minis while processing power can be up to 4 times that of the 16 bit mini.

6.4 INTERCONNECTION

A variety of methods have been developed for interconnecting processors. A number of these methods will be discussed below.

All of the mechanisms which we will discuss have been implemented and described in the literature and can therefore be considered feasible. Many of these mechanisms have also been commercially implemented. Examples of a number of commercial approaches will be given to assist the reader in understanding the approaches being presented and to provide a better overall picture of the current state of technology.*

Two broad categories of interconnection will be considered: shared links; and dedicated links.

6.4.1 Shared Links

Shared links allow two or more processors to communicate via a shared path. Shared links tend to be less expensive than dedicated links. Shared links also tend to be more flexible than dedicated links because the effective connectivity can be higher at a given cost. As with any shared system resource, however, contention for that resource can limit system performance.

*No endorsement of any of the products mentioned is intended or should be implied.

A shared link, since it is a common element, can be a point of single failure which could disable the entire system. Where availability requirements are high, redundancy is often employed to meet the requirements. Isolation must also be considered to ensure that a failure of the link cannot induce failures in the devices which share it.

Four types of shared links will be discussed below which include: shared memory; shared buses; shared serial lines; and circuit switches. Examples of commercially available systems will be given for each. Diagrams of the four types of shared link structures are shown in Figure 6-1.

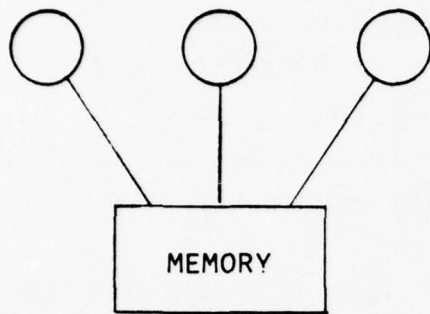
6.4.1.1 Shared Memory

Shared memory provides a powerful and flexible way of allowing processors to work together. Shared Memory can be interconnected with processors by time shared buses; circuit switches, or by multipointing.

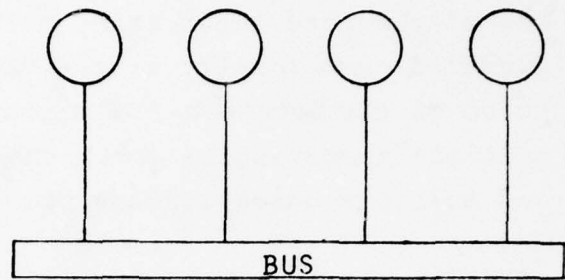
If I/O as well as memory is shared and control of the system is integrated the system is called a multiprocessor, otherwise the system is a multiple computer system with shared memory.

Shared memory, whether in a multiprocessor or multicomputer configuration, provides access for the processors to common data and programs. This sharing makes it possible in many systems to reduce the total memory available. However, unless the processors are slow relative to the memory, the potential exists for performance to be degraded because of contention for the shared memory. When memory contention will be a problem, it can often be overcome by the use of separate non-shared memories local to the processor or by segmenting the memory and providing separate memory controls so that a number of memory accesses can occur simultaneously. Both approaches increase the complexity and cost of the system, however.

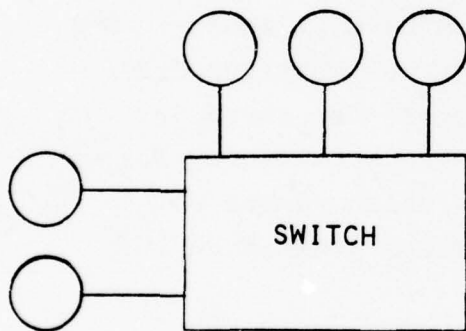
SHARED LINKS



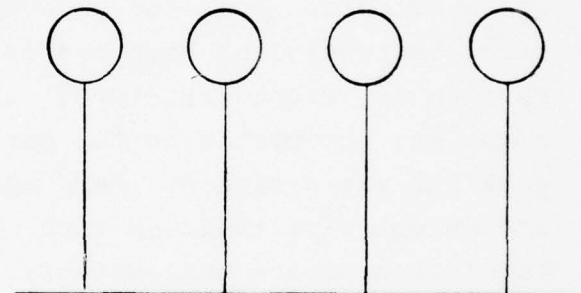
SHARED MEMORY



SHARED BUS



CIRCUIT SWITCH



SHARED SERIAL LINE

Figure 6-1. Shared Links

Shared memory systems are conceptually simple in that communication within the system is by simple memory access. In actual practice, however, these systems involve extremely complex interactions from both a hardware and a software standpoint. These interactions can make such systems difficult to utilize.

One of the more interesting multiprocessor systems which has been produced commercially is the PLURIBUS system developed by BB&N. Based on the Lockheed SUE minicomputer, the system featured multiple memory units (both shared and private), up to 14 processors, and multiple buses linking processors, memory modules and I/O.

PLURIBUS was designed to be fail-soft. The loss of a processor or a memory module results in a degradation in system performance rather than a loss of functionality.

Special hardware was provided in PLURIBUS to support a job queue which each processor accessed for its next assignment when it was available. Job steps, however, had to be kept short to assure timely servicing of high priority tasks since PLURIBUS did not have an interrupt structure. The requirement for short job steps was acceptable in the packet switching application for which PLURIBUS was designed. Many applications which cannot be conveniently partitioned into small job steps would find the PLURIBUS approach unacceptable.

For additional examples of multiprocessors and shared memory systems, the reader is referred to "Multiprocessor Organization - A Survey" by Philip H. Enslow, Jr., Computing Surveys, Vol. 9, No. 1, March, 1977.

6.4.1.2 Shared Buses

Time-shared buses can be used, as mentioned in the preceding section, to interconnect memory modules and processors in a shared memory or multiprocessor configuration. Time-shared buses can also be used to interconnect independent processing elements consisting of both processor and memory.

Buses can support high throughput with relatively simple digital logic. This is possible since multiple data and control paths can be provided.

With a bus as the interconnection mechanism, the hardware configuration can be modified easily by adding or removing processing elements. The addition of processors to a system, however, normally increases the utilization of the bus so that a point can be reached at which the addition of processors is counterproductive.

For dedicated systems, the bus can be designed with a bandwidth sufficient to support the maximum configuration thus allowing the system to be modularly expanded within design limits.

Two commercially available buses are worth noting as examples of two approaches to the use of buses.

The MultiComputer Adapter(MCA) is a device marketed by Data General which can interconnect up to 16 independent Data General processors. The bus is a time division multiplexed device with a bandwidth up to 8 megabits. Throughput is reduced as processors are added since additional time slots must be provided. The MCA is not an integral part of a normal Data General system. It can be purchased by the user as an option which interfaces with the normal I/O Bus and allows the user to configure a multiple computer system. Data General supplies I/O

driver software for MCA support but does not provide an integrated operating system for support of a multiple computer configuration.

Tandem's Dynabus, however, is an integrated part of the system and is not associated with the normal I/O bus. Figure 6-2 shows the basic structure of the Tandem system with the Dynabus. The Tandem system has been designed for uninterrupted processing. Processors (up to 16 may be included) are paired with the pairs having dual control of I/O devices. The operating system is designed around the system structure and supports interprocessor communication. Self checking routines are an integral part of the operating system which allows failed processors to be isolated quickly.

6.4.1.3 Shared Serial Lines

Shared serial lines are similar to shared buses except that a single path is provided for both data and control information.

Three types of control mechanisms are used with shared serial lines. The first is poll and select. In this approach one processor or control device is designated as the master. The others are slaves. The designation of master and slaves can be fixed or dynamic. The master controls allocation of the line by sending coded control messages to each processor in turn. If a processor has a message to send it does so when it is polled by the master device.

Polling introduces relatively high overhead and can result in significant time delays since a processor is required to wait its turn in the polling sequence even though it is the only processor wishing to transmit a message.

A polled line with a bandwidth of approximately 1 megabit is the standard interconnection for processors in Navy aircraft.

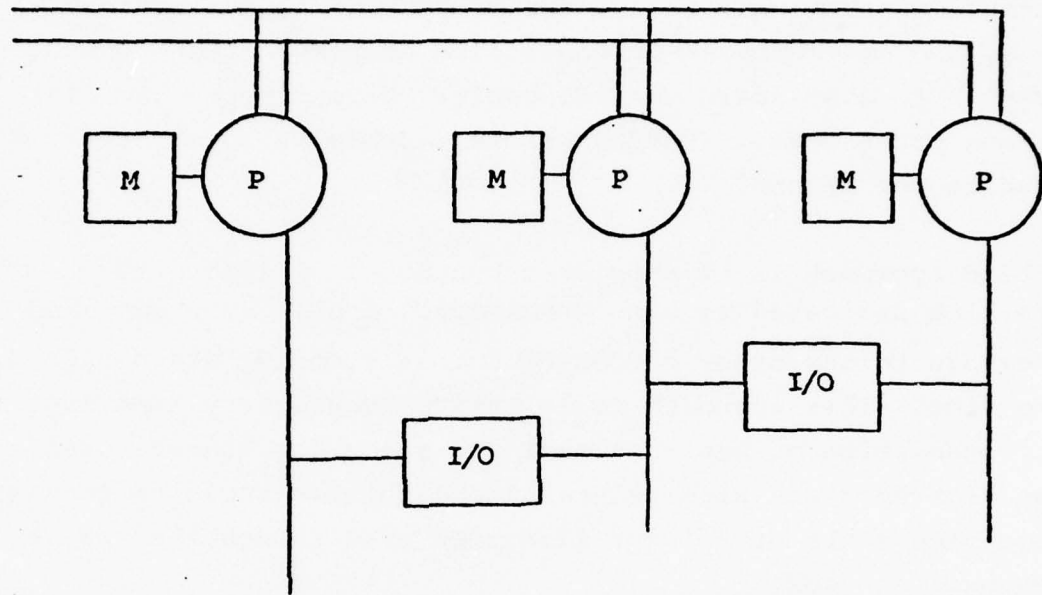


Figure 6-2. Tandem Configuration

A second approach, which reduces the overhead associated with polling, is a contention system. With contention each processor can bid for the line when it has a message to transmit. If the line is busy, the processor backs off and tries again at a later time. All processors are continually listening for messages with their addresses. A contention system for interconnecting heterogeneous computer is available from Network Systems Corporation. Called the HYPERCHANNEL, the system consists of microprocessor controlled adapters which connect each computer to a standard coaxial cable. Depending on the distances between processors, HYPERCHANNEL is capable of speeds up to 50 megabits per second.

A third approach is to provide a synchronized line with a time slot dedicated to each processor. A processor can send a message to any other processor but only during its assigned time slot. This approach could create unnecessary time lags in transmission of messages that are ready for transmission. The time slot approach also reduces the effective speed of transmission since time slots are always allocated even though they may be frequently unused.

6.4.1.4 Circuit Switch

A circuit switch is a set of input lines connected to a set of output lines by an array of cross-points. A circuit switch, as previously noted, can be used to interconnect an array of processors, memory modules and I/O devices in a multiprocessor or shared memory configuration. A circuit switch can also be used to interconnect a group of computers.

High throughput can often be achieved with a circuit switch since once the connection is made a dedicated path exists between the elements.

A variety of switch configurations have been developed including both blocking and non-blocking types. A non-blocking switch

such as a crossbar matrix can establish a connection between any two elements which are not currently connected to other elements. A conversation may be deferred because one of the parties is busy but not because of a limitation in the switch.

Blocking is possible, however, in a number of switch configurations. Blocking can occur when the number of paths through the switch is less than the maximum which could be needed. In a switch which allows blocking, a conversation may have to be deferred when both parties are free, because the switch is busy. A circuit switch which allows blocking is available from Data General for use in interconnecting Data General computers. Called the Data Distribution Network, the device allows up to 253 computers to interconnected. In the maximum configuration, 24 full-duplex conversations can proceed simultaneously at a 5 megabits per second rate.

6.4.2 Dedicated Links

Dedicated links are frequently used to interconnect computers. In a dedicated link configuration, a given link connects two devices. Sharing is not involved, so system performance cannot be degraded by contention for the link.

6.4.2.1 Dedicated Link Mechanisms

A number of mechanisms can be used to form dedicated links. The most commonly used dedicated link is a serial line. Serial lines can be used to interconnect processors which are collocated or processors which are separated by thousands of miles. Off-the-shelf devices are available from a number of manufacturers which allow speeds in the two megabits per second range for use in local environments and up to 50 kilobits per second when the telephone network is involved.

Parallel interfaces are also available for some systems which provide a channel to channel link that can transfer data from one computer to another at or near memory speed.

Dual port memory can also be used as a dedicated link with some systems. When used solely as a communications path, dual port memory provides a high speed link without the complexities normally associated with shared memory. Dual port memory is only applicable when the distance between processors is small.

Dual access peripherals can also be used as dedicated links. Dual access discs, for example, are available for use with a number of computers. Such dual access peripherals, however, are normally used as shared resources or in redundant processor configurations rather than as dedicated links.

6.4.2.2 Dedicated Link Structures

Computers interconnected by dedicated links can be arranged to form networks with a variety of topologies. Some of the more interesting structures are shown in Figure 6-3 and discussed below.

6.4.2.2.1 Fully Connected

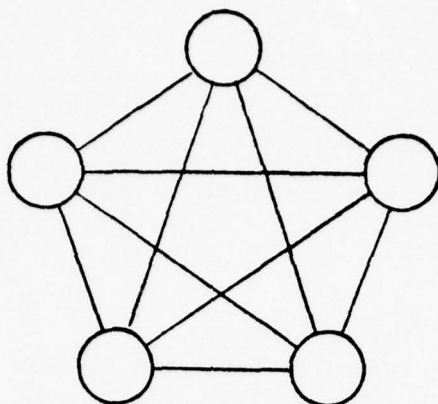
Fully connected networks provide dedicated links between each processing element and every other processing element. This approach allows direct point-to-point communication between any two processing elements. Full connectivity provides a high degree of flexibility in the way functions are allocated to processors since a dedicated link is provided for all conceivable data paths. Cost of a fully connected network can be quite large, however, because the number of links required for n processors is $\frac{n(n-1)}{2}$. The number of links grows as the square of the number of nodes to be connected.

6.4.2.2.2 Multiply Connected

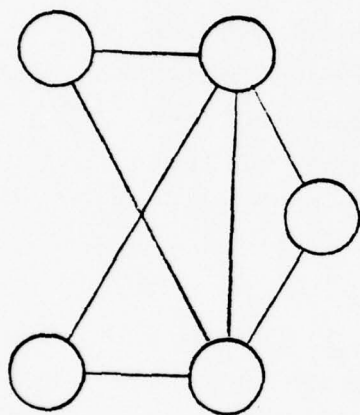
Multiply connected networks reduce the number of links required for full connectivity by eliminating those links which are not essential. Links can be eliminated if the corresponding logical data path does not exist or if data flow requirements allow messages to be forwarded by other processors.

While cost can be reduced dramatically, flexibility is likewise reduced relative to the fully connected network. Changes in the way functions are allocated to processors, for example, can result in substantial changes in the number and locations of links required.

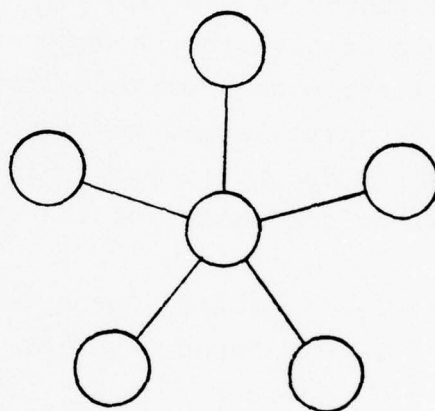
DEDICATED LINK STRUCTURES



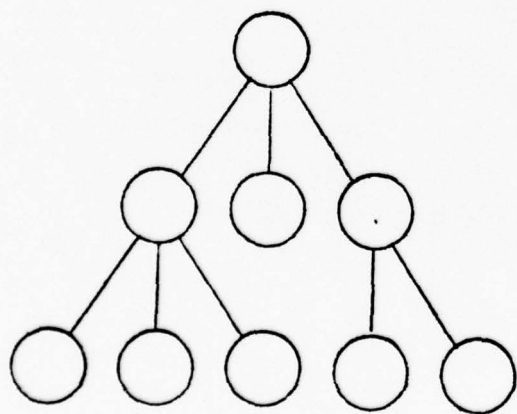
FULLY CONNECTED



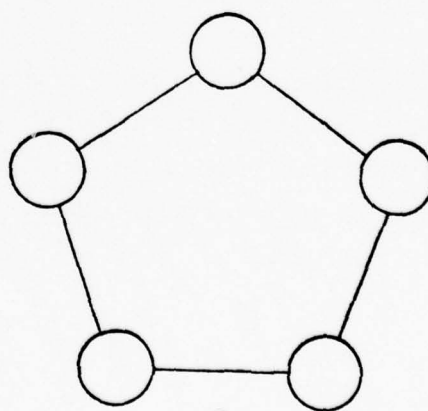
MULTIPLY CONNECTED



STAR



TREE



RING

Figure 6-3. Dedicated Link Structures
6-18

6.4.2.2.3 Star

A star network uses one node as a central communications element. All interprocessor communication is routed through the central node.

The star approach is included in the dedicated link structures since it normally utilizes the mechanisms associated with dedicated links. In many respects, however, it corresponds with the shared links since the central node can be thought of as a shared resource. As such the system is subject to the problems associated with contention for a shared resource and the potential for a total system outage due to the loss of the shared resource.

The star topology is also applicable to systems which constitute a two level hierarchy. In a two level hierarchy the lower level nodes do not normally communicate with each other. The central node would serve as the primary processor with the others subservient to it.

6.4.2.2.4 Tree

The two level hierarchy can be expanded to multiple levels forming what can be called a tree. Since many systems are logically hierarchical, the tree structure can be used in a variety of applications.

The tree structure is rigid, which limits its flexibility and a system so configured would be subject to failures of a hierarchy of critical nodes.

6.4.2.2.5 Ring

In a ring network each processing element is connected only to its immediate neighbors. Messages flow in one direction around the ring. Each node examines the messages and passes on those messages which are for another destination. Messages can be removed from the ring by the destination or can be allowed to continue around the ring to the sender as a way of verifying the integrity of the ring.

Like the star, the ring could be considered a shared link rather than a dedicated link. It is subject to the potential throughput limitations of a shared link. The ring, like the shared links, is also a potential failure point that can disable the entire system.

A ring could be constructed by interconnecting processors using standard dedicated link mechanisms. Such a configuration would be inefficient, however, since each processor would be interrupted to examine each message. Ring networks are therefore constructed with ring interfaces that handle the tasks associated with passing messages around the ring.

Ring networks exist primarily in a research environment.

6.5 LOGICAL STRUCTURE

The logical structure of a system is distinct from and frequently more significant than the physical structure. Much of the literature dealing with multiple processor systems has failed to properly distinguish between the logical and physical characteristics. This is due in part to the fact that particular logical structures lend themselves to some physical structures and not to others.

In the discussion which follows we will identify logical characteristics and structures and attempt to show how these can be related to the physical characteristics previously discussed.

6.5.1 Control Structure

Four types of control structure can be seen in multiple processor systems. These control structures are:

- . Independent
- . Distributed
- . Hierarchical
- . Integrated.

Independent control can be observed in a number of computer networks. Each processor manages its own set of tasks and its own resources. The network provides only a communication mechanism. Functions associated with the network are of little significance to the overall work of a given processor. Terminals connected to one processor may use another, but they do so just as if they were connected directly to the latter. There is no "support" provided by the local processor. A good example is the ARPANET.

Distributed control spreads the management of a common set of tasks and resources among a number of processors. Distributed control is normally a feature of a system with a shared link which provides full logical connectivity.

Hierarchical control centralizes control in layers. Lower level processors are subservient to higher level processors which may in turn be subservient to one or more additional levels. Hierarchical control can be implemented with a network that provides a hierarchical interconnection structure such as the tree described in Section 6.4.2.2.4. Shared link structures, normally associated with distributed control, can also support hierarchical control.

Integrated control is a characteristic of true multiprocessors. PLURIBUS, for example (See Section 6.4.1.1), featured an integrated operating system with special hardware to assist in integrated task management.

6.5.2 Degree of Cooperation Among Processors

The degree of cooperation among processors in multiple computer systems ranges from essentially none to nearly total. The degree of cooperation required influences the selection of interconnection mechanisms. The degree of cooperation which is possible is determined by the interconnection and the control structure. However, the degree of cooperation actually utilized may be significantly below that which is theoretically possible.

In general, as the degree of cooperation is increased, the potential for load balancing and optimization of resources

is also increased. Increased cooperation, therefore, provides an increased potential for improved system performance. This is counterbalanced, however, by a corresponding increase in system complexity and interprocessor communication overhead.

6.5.3 Assignment of Functions

Functions to be performed in a multiple processor system may be assigned to processors either statically or dynamically.

Dynamic assignment of functions has theoretical advantages in terms of throughput but increases system complexity. Dynamic assignment is normally reserved for multiprocessors.

Static assignment of functions is normally used in multi-computer systems. Each processor is assigned a specific task or group of tasks which it always performs except when the system is reconfigured. Reconfiguration could take place because of relatively long term changes in system requirements, or to work around a failed piece of equipment.

A number of approaches can be used to determine the most effective static assignment of functions. By grouping functions in processors so that functions using the same devices or data are together and attempting to optimize processor and memory utilizations we can effectively minimize the number of processors required. If processors are to be minimized, however, functional allocation must be changed from system to system.

Another approach would attempt to maintain the same functional allocation from system to system. This operation simplifies system generation but is less effective in minimizing the amount of hardware required.

The previous chapter presented a broad range of characteristics which can be combined to form a variety of system architectures. If all the combinations were tabulated, an extremely long list would result. In this section we will reduce the list to four architectures which will be analyzed and evaluated in subsequent chapters.

7.1 LOGICAL AND PHYSICAL CHARACTERISTICS

We will review the logical and physical characteristics presented in the previous chapter and select those characteristics which will best meet the needs of the VTS Processing/Display Subsystem. The selection will be based on U. S. Coast Guard requirements and the preliminary design study presented in the preceding chapters.

7.1.1 Processors

Four classes of processors were discussed in the previous chapter. Included were:

- . 8 Bit Micros
- . 16 Bit Micros/Small Minis
- . Standard Minis
- . 32 Bit Minis

Eight bit microprocessors could be used as the building blocks for VTS. However, from the processing requirements developed in Section 3 and summarized in Figure 3-27, it appears that the use of eight bit micros would result in substantial fragmentation of the processing in the large VTS configurations.

A more serious problem for a system such as VTS could be the limited software support and relatively primitive development tools

available for use with eight bit micros. VTS software must be flexible and easily modified. Requirements may change significantly over the life of the system making software development facilities extremely important.

To overcome the lack of software, integrated hardware and peripheral support would require the development of a microprocessor based multiple processor system with full hardware and software support. This would require considerably more development than is anticipated by the U. S. Coast Guard.

The 16 bit micro/small mini class of processors seems to be a reasonable choice for the low end of the processor spectrum to be considered for VTS since the basic processing power matches the processing required for the Class A, Level 1 system. The limited memory and the lack of floating point hardware could prove to be a disadvantage, but this can be determined during the evaluation of the selected architectures.

Standard minis should also be given further consideration. The increased processing power and the flexibility to add large memory, if required, could prove advantageous. The systems could be more costly than with the smaller processors, however, since a single standard mini provides more processing power than is required for the smallest VTS system.

A thirty-two bit mini would provide sufficient processing power for even the maximum VTS configuration. Since hardware modularity is a design goal for the VTS Processing/Display Subsystem, 32 bit minis do not seem appropriate based on current estimates of processing loads.

Either the 32 bit mini or the 8 bit micro can be examined more closely at a later time. If an architecture based on small processors is

selected, the use of 8 bit micros can be reviewed. If the larger processors prove the most attractive, the 32 bit minis can be given further consideration.

7.1.2 Interconnections

Both shared and dedicated links should be considered for the candidate architectures.

7.1.2.1 Shared Links

The VTS system will include processors which are closely colocated. It will also include a number of processors associated with the display stations which will normally be located in the adjacent room, but would be separated from the other processors by distances on the order of 100 feet.

Since shared memory configurations have not normally been designed for such distances, shared memory is the least desirable of the shared links considered.

The other shared links provide a shared communications mechanism that could function in conjunction with the other physical and logical approaches. Since any one of the other shared links could be used we can consider them as a group and defer a selection of the particular mechanism.

7.1.2.1 Dedicated Links

Of the dedicated link mechanisms reviewed in Chapter 6, dual port memory is the least desirable because of distances involved. Serial lines are the most widely used and will be assumed unless further analysis indicates a need for the higher throughput which could be achieved with parallel interfaces.

From the dedicated link structures we can eliminate the ring structure, since it is still experimental. The fully connected structure is

unrealistic for the number of processors required for the large scale VTS system (12 display station processors are required).

The star and tree structures are dependent on one or more critical nodes. For reliability, these nodes would be duplicated forming a modified star or tree. Such a configuration would also be considered multiply connected. Thus, some form of multiply connected structure is the only real alternative.

7.1.3 Control Structure

Control structures will be selected in conjunction with the physical structures. Independent control is not applicable since the processing elements must work together in a VTS system. Integrated control is also not applicable since the selected physical structures cannot support it.

In a multiple computer system, the distinction between hierarchical and distributed control becomes essentially a matter of degree.

7.1.4 Degree of Cooperation

In all the architectures which we will investigate, the degree of cooperation among processors will be moderate.

7.1.5 Assignment of Functions

Assignment of functions will be essentially static in all the architectures considered. The two general approaches to static assignment discussed in Section 6.5.3 will both be considered.

7.2 DEVELOPMENT OF CANDIDATE ARCHITECTURES

In the preceding section we selected two classes of processors, two types of interconnection, and two approaches to functional allocation.

For the first architecture to be considered we will use the 16 bit micro/small mini class of processor. Since more processors will be required when the less powerful processors are selected, we will use a shared link to interconnect the processors. To prevent undue hardware complexity we will allocate functions so as to minimize the number of processors.

For the second architecture we will use the larger processors while leaving the form of functional allocation and the approach to interconnection unchanged from the first architecture.

Larger processors, and the corresponding lesser number of processors, makes the use of a dedicated link approach more attractive than it would have been with a larger number of processors. The third architecture will be identical to the second except for the use of a dedicated link interconnection.

For the fourth architecture we will use the second approach to functional allocation. Functions will be allocated to processors so the same allocation can be used for all classes and levels of systems. We will assume that a family of processors can be used so that the capability of a processor can be matched with the functions allocated to it. A shared link will be assumed since it provides a higher degree of flexibility.

The four architectures selected include both types of processors, both methods of interconnection, and both approaches to functional allocation. Although all possible combinations have not been considered, the analysis of the four selected architectures should provide sufficient insight to judge other possible combinations as well.

The set of architectures to be considered represents a set of complex systems. These candidate architectures were selected from various possible architectures based on some coarse structural criteria. This selection process was summarized in Section 7.

The end result of Phase I is to select two feasible architectures from the viable candidates chosen under Task 7. Each candidate is composed of information processing and communications subsystems. Thus, attention must be paid not only to the characteristics of individual elements but also to their interconnections and interactions. The elements of these systems will operate in a highly integrated and interdependent manner. The degree of integration and interdependency is related to the architecture and the combination of elements which compose it.

The feasible architecture selection process requires a set of evaluation criteria which can readily identify the two 'best' architectures from among the viable candidates. Eight categories of evaluation criteria have been identified. Each category will be applied to both hardware and software components. Each category is composed of a number of factors.

The method of analysis and evaluation is straightforward. Each criterion is given a weight relative to other criteria. This value is divided (in some ratio) between hardware and software components. Each architecture receives a figure-of-merit based on the sum of its weighted scores for all criteria. The two architectures with best figures-of-merit are deemed to be the two best architectures.

Some of the criteria which are discussed in the following sections cannot be fully evaluated at this time since they relate to the specific hardware and software which would be used in an actual system. Such criteria are included here for completeness and will be used subjectively where differences among candidate architectures can be discerned.

8.1 SIMPLICITY

While a degree of complexity is necessary for all real systems, unnecessary complexity makes a system more difficult to build, operate and maintain. Simplicity is therefore an important criterion in the evaluation of alternative system architectures.

8.1.1 Hardware Factors

Number of Hardware Components

A distributed processing system is made up of a number of hardware components such as processors and interprocessor links. In measuring simplicity, the total number of such identifiable components should be considered with a higher score being given for a lesser total number of components.

The number of different types of components should also be considered with a higher score given for a lesser number of different types of components. As an example, consider two systems, one of which has ten identical processors, the other has four processors, each of a different type. The system with the ten identical processors would receive a lower score based on the total number of components but a higher score based on the number of different components.

Structural Complexity

Structural Complexity is a function of the way in which system components are interconnected. In a multicomputer system evaluation it is important to consider both the number of interconnection paths and the complexity of the structure. The more paths (i.e., interconnection devices) required the lower the score. This can be counterbalanced, however, by the complexity of the structure. For example, the fully connected network requires more interconnection paths than the other structures considered (see Section 6.4.2.2.1). It is, however, conceptually less complex than the multiply connected network because of the completeness of the structure.

Redundancy

Redundancy is usually needed in systems such as the VTS Processing/Display Subsystem which require high availability. Redundancy, by its very nature increases the complexity of a system. For the evaluation we will consider the effect that the need for redundancy has on the structure of the system. Some system architectures can accommodate additional processors easily to provide the redundancy that may be needed without substantially increasing system complexity. Other architectures become much more complex if redundancy is added.

8.1.2 Software Factors

Number of Software Components

The number of software components is a significant factor in determining the simplicity of a computer system. The number of processes, programs, and data bases should be considered. A higher rating is given for a smaller number of software components.

Structural Complexity

Structural complexity relates to the number of communications paths between the software components and the structure formed by the communications paths. A higher rating is given for a lesser number of communications paths. A higher rating is also given for a less complex structure.

Redundancy

Redundant hardware and backup data bases and software must often be provided to meet high availability requirements. Redundancy and special backup software, of necessity, adds to the complexity of a software system. Architectures which can support the need for high availability in a straightforward manner should receive a higher rating.

Duplication of data and software, which is required by the architecture, and not by the need for availability can be particularly troublesome in a computer system, increasing the difficulty in implementing and validating the system software. An architecture which requires several specialized copies of a data base to enable it to function should receive a lower score.

8.2 FEASIBILITY

The feasibility of an architecture is related to its suitability and capability to provide a given level of performance within current and near-term systems technology. Constraints on feasibility include design and development strategies, costs, manpower and time resources, and testing considerations.

8.2.1 Hardware Factors

Off-the-Shelf Availability

This factor relates to the availability of basic units of the architecture which have the desired performance characteristics. These include the central processors, interprocessor communications units, graphics terminals and disc units. A higher rating is given for each type of component if the component is available from two or more manufacturers and is attachable to two or more minicomputer vendor's products.

Technological Feasibility

If all hardware cannot be obtained off-the-shelf it is important to assess the technological feasibility of the hardware to be developed. A particularly low score should be given an architecture which requires the solution of significant technological problems. An architecture which depended on technology which exceeds the state-of-the-art in hardware design would be rejected immediately.

Interface Requirements

This factor relates to the ease of interfacing different units and the number of units which may be supported by a given component. A lower rating is given if special interfaces must be developed or additional components are required to support the interfaces. A lower rating will also be given if special software considerations are required to handle the interface.

8.2.2 Software Factors

Off-the-Shelf Availability

This factor relates to the availability of standard system software which can satisfy the performance and support requirements. Included in this group are the operating system, file management system and high order systems implementation language. The operating system includes device, input/output and file-management services as well as support for high order programming languages. A higher rating is given for availability of system software from at least two minicomputer vendors meeting the hardware criterion above.

Technological Feasibility

If system software is not available off-the-shelf, the technological feasibility of the required system software should be carefully considered. If totally new, unproven software concepts must be used to allow the system to function, the architecture requiring such concepts would be rejected. However, multi-computer system software involves state-of-the-art design and depends to some degree on concepts and techniques which are only partially proven. The risk associated should be considered. A higher score should be given for a lower risk, i.e., for less dependence on unproven concepts.

Software Development/Implementation Time

This factor is a subjective estimate of the time required to develop and implement the applications software. It is related to the structural complexity of the system architecture. This estimate is subjective since a detailed system/subsystem specification has not been prepared. A higher rating is given for a lower estimate of software development/implementation time.

System Generation/Modification Time/Effort

This factor relates to the time and effort necessary to install a VTS system. Prior to system generation, waterway traffic and characteristics studies must be performed in order to establish several system parameters. Appropriate modules must be identified and selected for the system. Specific modules for customized functions must be developed and tested. Waterway maps must be generated for the display stations based on the number of sectors. Once a system is installed, modifications may be required due to traffic growth, new Coast Guard regulations or changing waterway characteristics. A higher rating will be given for a lower estimate of the time and effort necessary to generate and modify a VTS based on the candidate architecture.

8.3 MODULARITY/FLEXIBILITY

Built-in flexibility is the ability of a system to handle different logical situations. In the VTS project, flexibility is the architectural response to the varying waterway characteristics, traffic loads, changing regulations and local customs. Built-in flexibility, a design requirement, may increase the system complexity since it may require an increased number of communications and decision paths in the system.

Modularity implies a division of a system into a number of modules which can be thought of as separable components. These modules have limited interfaces to other modules. Systems can be divided into modules in many ways: frequency of operational use, module size, or machine dependency. Modularity can provide both benefits and disadvantages. The degree of the effect relative to system or program attributes is the focus of this set of factors.

8.3.1 Hardware Factors

Excess Capacity

This factor relates to the potential of the system to handle varying and possibly increasing loads without modification. Two questions must be considered: What increase in utilization of a subsystem can be sustained and how does this increase impact the performance of the system? A higher rating is given to excess capacity which can result in greater utilization without impacting performance.

Switchability of Data Control Paths

This factor relates to the survivability of the system and the ability to handle varying traffic loading conditions across the waterway. Alternate paths for data/control messages allow the system to survive component failures. A higher rating is given to a system which has multiple data/control paths to handle total message traffic load.

AD-A061 307

INTERNATIONAL COMPUTING CO BETHESDA MD

F/G 15/5

PRELIMINARY DESIGN STUDY FOR VTS PROCESSING/DISPLAY SUBSYSTEM.(U)

JUN 78 C C HENSON, R A CLEAVER, S H KAISLER

DOT-CG-81-78-1833

USCG-D-66-78

NL

UNCLASSIFIED

3 OF 4
AD
A061307



Adaptability

This factor relates to open-ended flexibility, i.e., the degree to which interfaces are designed to specifically allow future change with relative ease. The number of linkages between modules is an indicator of the number of places where changes can be "plugged in" without major disturbance in the existing system. A higher rating is given for a higher degree of adaptability.

Generality of Architecture

This factor is related to the degree to which a system is applicable in different environments. A higher rating is given to an architecture which has a greater degree of applicability over a variety of waterway characteristics.

8.3.2 Software Factors

Parameterization

Parameterization of software contributes significantly to the ability of the system to deal with changing conditions and thus increases flexibility. Parameterization makes it easier to adapt the system for use in different ports and waterways. It also allows functions to be provided to control parameters of operation, thus providing flexibility in adapting to changing needs and conditions.

Allocation of Functions

The degree of flexibility provided by the architecture in the assignment of functions is important in creating systems for a variety of environments. Flexibility is also needed to allow the system to be easily reconfigured in the face of a failure of a portion of the system. A lower rating should be given to an architecture which limits the flexibility to assign functions as needed.

Evolutionary Project Development

This factor is related to the degree to which the applications software can be developed in an incremental fashion. A flexible

system is one which can evolve to the maximum operational configuration without requiring massive software or structural modifications. A higher rating is given to a system for which a subjective estimate predicts a high degree of evolutionary development.

Adaptability

As with hardware, software adaptability is related to the degree of open-ended flexibility. The number of software interfaces exist to a much greater degree and apply not only to code segments but also data structures and data bases. A higher rating is given to an architecture where a subjective estimate predicts greater flexibility.

Generality of Software Architecture

This factor is related to the degree to which the applications software applies to different waterway environments. Conversely, it is a subjective estimate of the implications of architecture on the number of customized modules which need to be developed for a given VTS installation. A higher rating is given to an architecture which is estimated to reduce the number of customized modules.

8.4 MAINTAINABILITY

System Maintainability is a measure of the difficulty of preventive and remedial maintenance for both hardware and software components. Associated with this concept is the idea of repairability, i.e., that a failed subsystem can be restored to operable condition within a specified time limit. Another allied concept is serviceability, i.e., that the problem can be analyzed and the solution defined. These three concepts rely upon several different factors for a maximum rating.

8.4.1 Hardware Factors

Design Adequacy

This factor is related to the degree of maintainability that has been incorporated in the design. Features include modularity and standardization of parts. Design adequacy is basically a subjective measure until the system is built. A higher rating is given for a greater degree of design adequacy.

Diagnostic Capability

The ability of operations and maintenance personnel to locate a problem is an important factor in system maintainability. A system could be highly reliable, thus requiring infrequent maintenance, but difficult to maintain if diagnosis of a failure were difficult when it occurred. A higher rating is given to an architecture which can be easily diagnosed.

Personnel

A key element in system maintenance is often the repair staff. Maintenance of some systems requires a maintenance staff with a high skill level, strong motivation and specialized experience. A higher rating should be given to a system which does not require a highly skilled maintenance staff.

Support Facilities

This factor relates to the ease with which a failed system may be repaired. Tools may be needed locally. A spare parts inventory emphasizing the 'remove and plug in' approach to repair is often desirable although the size and breadth of the inventory may vary depending on the architecture. A higher rating is given for an architecture which minimizes the need for support facilities and spare parts inventories.

Inherent Reliability

The mean time before failure (MTBF) and the mean time to repair (MTTR) are measures of the inherent reliability of system components. The use of inherently reliable components reduces the need for maintenance. Although specific hardware will not be selected at this time, the inherent reliability of various types of equipment can be considered in evaluating architectures. A lower score should be given based on the number of inherently less reliable components such as mechanical devices (e.g., discs).

Standardization

This factor relates to the repairability of equipment through similarity of components. By standardizing the component set, one minimizes the size of the spare parts inventory and limits the number of diagnostics required to detect a problem. A higher rating is given for a greater degree of standardization.

8.4.2 Software Factors

Design Adequacy

This factor relates to the use of good design techniques and management practices which ensure a structured, comprehensible system. A higher rating is given to a system which is amenable to these techniques.

Built-in Diagnostics

This factor relates to the facilities available for diagnosing a problem in the applications software. Good error detection and correction algorithms should be built into the software. A higher rating is given for more detailed and obvious error detection and correction facilities.

Support Personnel

This factor relates to the ease with which software errors can be detected and corrected, and to the ease of incorporating additional functions and features in the software. A higher rating is given for a system which does not require highly skilled support personnel for maintenance. Since the USCG does not anticipate on-site software support, a lower rating should be given for a system which requires such support.

Support Facilities

This factor relates to the ease with which problem solutions can be tested to ensure correctness. Usage of the simulation feature in the VTS provides a good testbed facility for module checkout. A higher rating is given for the degree to which facilities are available to test and install problem solutions.

Standardization

This factor relates to the standardization of procedure and data structures which makes it easy to assemble and link modules for any VTS installation. A higher rating is given for an estimate of a greater proportion of standardized code and data skeletons.

Error Detection and Correction

This factor relates to the ability of the system to maintain its integrity in the face of software or data base faults. Timely detection of errors is essential to system integrity. Correction procedures may be automatic (as in communications systems) or require support personnel. A higher rating is given to architectures which possess the inherent capability to support a greater degree of error detection and correction procedures.

8.5 EXPANDABILITY

The expandability of an architecture is related to its growth capability in both performance and data storage capacity and its ability to add new functions or features. An aspect of expandability is evolution which is the design objective of spreading out necessary changes over a relatively long period of time so that the rate of change as a function of time is low. Evolutionary growth must maintain the stability of the system and thus implies a stepwise refinement of system capabilities. Since the VTS design must encompass a wide range of capabilities and will be expected to support functions which are not yet fully defined, it is particularly important that the architecture selected be expandable.

8.5.1 Hardware Factors

Effort Required to Add Capability

This factor relates to the ease and cost with which an architecture can be expanded. Changes should minimize the disturbance to the system operation and affect as few modules as possible. A higher rating is given for a higher subjective estimate of the ease with which an architecture accommodates change.

Replace Modules as Technology Changes

This factor relates to extending the lifetime of the architecture by making appropriate usage of improvements in technology. Over the lifetime of a VTS installation, technological improvements in various areas of hardware can be expected. A higher rating is given to an architecture which can easily accommodate changes due to technological advances by replacing or upgrading subsystem components.

Adding or Upgrading Processors

This factor relates to the ease with which system performance can be improved by adding processors or upgrading to larger, faster processors within a family of processors. A higher rating is given to an architecture which can easily accommodate additional or more powerful processors.

8.5.2 Software Factors

Capacity

This factor relates to the ability of the software to handle an increasing workload. An impact of increasing workload or changing geographical constraints is a change in main memory or data base storage requirements. A higher rating is given to an architecture which minimizes the adjustments to memory/data base because of changes in functional allocations.

Effort Required to Add/Subtract Software Modules

This factor relates to the adaptability of the baseline architecture to the constraints of a particular waterway environment. Waterway characteristics and local regulations and traditions impose the need for accommodating customized modules. A higher rating is given to an architecture which minimizes the effort necessary to add or subtract functions and features to/from the baseline capabilities.

Effort to Replace Modules

This factor relates to the evolutionary growth of the applications software in response to changing system requirements. The transition in mode or level of a VTS installation may require a new system generation or just the replacement of functional modules. A higher rating is given to an architecture which emphasizes a cost effective approach to system transition.

8.6 RELIABILITY

System reliability is a complex measure of the probability that a system will perform satisfactorily (with no malfunctions) for at least a given time interval, when used under stated conditions. The concept of reliability includes actual operating time, down time and system effectiveness. This last concept further involves mission reliability, operational readiness and design adequacy. Software reliability further includes veracity and viability. Veracity is the degree to which the software represents the real world. Viability is the adequacy and accuracy with which the software continues to meet system requirements in unusual situations or in the face of error faults.

8.6.1 Hardware Factors

Availability

This factor relates to the instantaneous ability of the system to satisfactorily meet the operational requirements. In general, availability relates to both operational as well as useful availability. Basic availability is defined to be the ratio of operational time to total time. A higher rating is given for an architecture which assures a high availability.

Duration of Outages

Availability should not be considered alone, however, since the duration of system outages can be significant and is not reflected by the availability measure itself. The calculated availability for two systems might be the same if one of the systems failed frequently but was out of service for a very short time and the other system failed infrequently but was out of service for a much longer time when a failure did occur. A higher rating will be given for a lower estimate of average down time.

Recovery Time

This factor relates to the ability of the architecture to select an alternate path when recovering from a hardware malfunction. A higher rating is given for a lower subjective estimate of the recovery time.

Degree of Redundancy

The availability of a system depends on its ability to work around malfunctions or module losses. By providing redundant elements, the architecture increases the probability that alternate paths will be available. A higher rating is given to an architecture which possesses a higher degree of redundancy across all critical modules and which avoids single-point failure situations.

Degradability

This factor relates to the ability of the system to perform its tasks in the event of one or more hardware malfunctions. As malfunctions accumulate, system performance may deteriorate to the point that execution speed of some operational tasks may be reduced. A higher rating is given to an architecture which can degrade gracefully and maintain essential functions in the face of multiple failures.

Fault Detection/Correction

This factor relates to the ability of the system to preserve the availability after experiencing certain classes of faults. These faults can be detected and corrected at the local module level without requiring a change in system control. A higher rating is given to architectures emphasizing greater fault detection and correction.

8.6.2 Software Factors

Verification/Validation (V&V)

These factors relate to the ability to test and guarantee that the software is operational and will perform satisfactorily within the expected workload. A higher rating is given to architectures which facilitate the use of strong V & V procedures in the design and implementation of the software.

Recovery Time

This factor relates to the ability of the system software to divert system control and data flow around a malfunctioning module. Recovery is a function of a setup time to install the capability on a spare module or rebuild a system table. A higher rating is given to an architecture which minimizes recovery time.

Degradability

This factor relates to the capability of the software to adjust the workload level and distribution in response to module malfunctions. The system software may be forced to isolate hardware modules or reduce execution speed for some applications tasks (load-shedding) in order to maintain satisfactory performance for essential tasks. A higher rating is given to an architecture which emphasizes system reliability in the face of multiple malfunctions.

Effectiveness Under Unusual Load Conditions

This factor is complementary to the one above in that it relates to the ability of the system software to distribute tasks during unusual load conditions in order to maintain a satisfactory performance level. A higher rating is given to an architecture which is well behaved under unusual load conditions.

Error Detection/Correction

This factor relates to the ability to detect and correct local faults without changing system control or data flow. A higher rating is given to an architecture which emphasizes this feature.

8.7 COST

System costs provide a common baseline for comparing systems. Costs result from a number of factors and an attempt must be made to consider all relevant life cycle costs in evaluating the architectures. At this stage of the design many of these costs cannot be quantified but will be considered qualitatively in the evaluation.

8.7.1 Hardware/Software Factors

Development Costs

This factor relates to the costs expended in the requirements analysis, design, and development of the hardware/software system. A longer design period may engender greater development costs but this fact must be weighed against the probability of obtaining a better system, easier implementation and a more modular system.

Implementation Costs

This factor relates to the coding and testing of software in conjunction with a testbed facility. Usage of structured programming methods, chief programmer teams, program test data generators, etc., as well as systems simulation may significantly improve the quality of the hardware/software product.

Installation Costs

This factor relates to the integration of hardware and software at a VTS installation. It also includes complete system checkout and acceptance tests.

Maintenance Costs (Contract/Inhouse)

This factor relates to the cost of maintaining the hardware and software system. Maintenance costs are often a significant part of the overall life cycle cost. Maintenance strategies, however, have not been established and maintenance costs are a function of a number of factors other than the architecture selected. A relative factor based on the maintainability criterion will therefore be included in the overall cost factor.

Spare Parts Inventory

This factor relates to the cost of developing a supply of commonly used and easily replaceable parts at each VTS installation. Availability of spare parts means quicker recovery and repair in the event of a hardware failure. The number of spare parts required will depend on the size of the VTS installation. The number of spare parts required will also depend on the architecture selected and the relative need for rapid repair.

Conversion

This factor relates to the modification of any vendor-supplied hardware or software necessary during system development. A high probability exists, given the requirement for off-the-shelf equipment, that some modifications may be needed to support overall VTS functional requirements.

Useful Life (Depreciation/Amortization)

This factor is used to judge the life cycle expectations of the architecture versus future cost benefits.

Availability of Lease/Purchase/Rental Options

This factor relates to the acquisition method for selecting hardware modules. Over the life cycle of the VTS installation, lease of certain components (tape drives, printers, etc.) may be more

cost effective than outright purchase. This would allow the user to take advantage of advances in electromechanical systems technology.

Training Costs

This factor relates to the cost of training operators on the combined hardware/software system. It is a continuing cost since USCG personnel are not normally assigned permanently to an installation.

Requirements Change Cost

This factor relates to system modification and enhancements as the waterway characteristics or VTS installation characteristics change. The system as currently specified encompasses a broad range of requirements which will change in response to new technology, operational experience and the development of new techniques for providing service to the maritime community.

Documentation Costs

This factor relates to the cost of developing and maintaining system documentation for all VTS installations. Both baseline documentation and documentation for customized modules must be maintained. As requirements or operational procedures change, different manuals must be updated.

Systems Generation Studies

Prior to installation of a VTS, a major study of waterway characteristics and traffic loads must be performed to determine system parameters. Sector maps must be digitized. Local waterway customs and traditions must be analyzed to determine their impact upon the baseline system.

Operations Costs

This factor relates to non-ADP costs for the operation of a VTS such as supplies, electricity, etc. The impact of the VTS architecture and size on these costs must be carefully weighed.

Personnel

This factor relates to the manpower costs for a given VTS system. The number of watchstanders will be fixed by the installation. Additional personnel for maintenance and computer operations must be considered particularly for the larger installations.

8.8 PERFORMANCE

System performance is the ability to accomplish the given workload within the designated time constraints. Performance can be evaluated for individual components as well as for the overall system.

8.8.1 Hardware Factors

The hardware components of a VTS must be evaluated in the context of the whole system. In defining the architectures, individual factors such as the following have been taken into consideration:

- . Response to Interrupts
- . Central Processing Unit Speed
- . Disk Speed
- . Interconnection Bus Bandwidth
- . Main Memory Required and the Maximum Available
- . Mass Storage
- . Subsystem Utilization

These factors must be weighed together to estimate their impact on the overall system architecture. The tradeoffs between processing time and memory must be carefully considered within the context of subsystem utilization. The impact of peak loads on system performance is crucial to the determination of excess capacity and its location in the architecture. In summary, a balance must be achieved among the performance factors within a candidate architecture which does not significantly increase cost.

8.8.2 Software Factors

In a similar manner to hardware, a balance must be struck among different performance factors for software which does not

significantly increase system cost. The evaluation of these factors is very subjective because a detailed software design (or architecture) has not yet been performed. Among the factors to be evaluated are:

- . Response Time
- . Overhead Processing
- . Data Base Retrieval
- . Communications Load
- . Background Processing Time/Load
- . Excess Capacity
- . Data Transformation Time
- . System Efficiency
- . Operational Ease

The first candidate architecture to be considered is based on using small processors connected by a high speed shared link. System functions are distributed among the processors to form what is commonly called a distributed function multiple processor.

9.1 PHYSICAL STRUCTURE

Physical structure of an architecture is primarily determined by the type of processors and the manner in which the processors are interconnected. These two aspects are discussed below.

9.1.1 The Processors

Relatively small low cost processors were selected for this architecture. These processors were assumed to be capable of performing approximately 600,000 instruction cycles per second. Up to 128k bytes of memory could be supported by the processor. Floating point hardware is normally not available for these processors.

These characteristics are representative of a number of 16 bit microcomputers or small minis which are commercially available.

Eight bit microprocessors were considered for this type of architecture but were rejected because of significantly lower processing power. Based on the processing load estimates given in Chapter 3 the lower processing power of eight bit micros would lead to extensive fragmentation in the processing and unnecessary complexity in the system.

More powerful processors will be considered in subsequent chapters.

9.1.2 Interconnection

Processors will be interconnected by a shared link which will be replicated for reliability. The shared link could be an 8 or 16 bit bus or a shared serial line. The required throughput for this shared link will be discussed in Section 9.3.2.

9.2 LOGICAL STRUCTURE

Logically the system consists of a number of small processors each performing an assigned function or group of functions.

In the interest of logical simplicity it would be desirable to have a one to one mapping of processors and functions. A one to one mapping attempts to make each processor a separate entity and minimizes the interdependence of processors. In general, however, a one to one mapping is impractical since it would require either a wide variety of processors or a significant amount of wasted processing power.

For this architecture we have chosen to assign more than one processor to a function when the processing load for that function exceeds the power of a single processor. Likewise, functions are combined when a single processor is capable of handling more than one function.

In combining functions we have attempted to group together functions that make use of the same devices or data. This approach minimizes the interdependence of processors and reduces the load on the interprocessor links.

Function allocations common to all three cases which follow are:

- . Exception functions (as defined in section 3) are included in the Operating System (OS) nucleus for each processor
- . The relative position function is performed by the Display Station Processor (DSP)

9.3 CLASS C, LEVEL 4 SYSTEM

In this section a preliminary design is presented for a Class C, Level 4 system capable of handling 900 identified vessels. The structure of this system is shown in Figure 9-1.

9.3.1 Processor Sizing and Utilization

Processors have been assigned for each of the major system functions. In making these allocations we have attempted to keep CPU utilization in the neighborhood of 50%. CPU utilization is kept relatively low to prevent difficulties in meeting response time requirements and to allow for some margin of error in the processing loads developed in Section 3.

The following sections describe the processors which will be required for the 900 ship, Class C, Level 4 system. Memory sizes and CPU utilizations have been computed for each of the processors and are summarized in Figure 9-2.

9.3.1.1 Position Input Processors

Eight processors will be required to support the processing associated with radar input or other position input sensors. Each processor would be assigned to serve one radar unit. The processor could be considered an extension of the radar unit since its primary function is to convert data from the radar coordinates to system coordinates and provide an interface between the radar units and the other system processors.

Memory sizes and processor utilization for the position input processor are tabulated below. Processing and memory sizing assumes a maximum of 512 tracks of which 256 will be vessels requiring coordinate transformation.

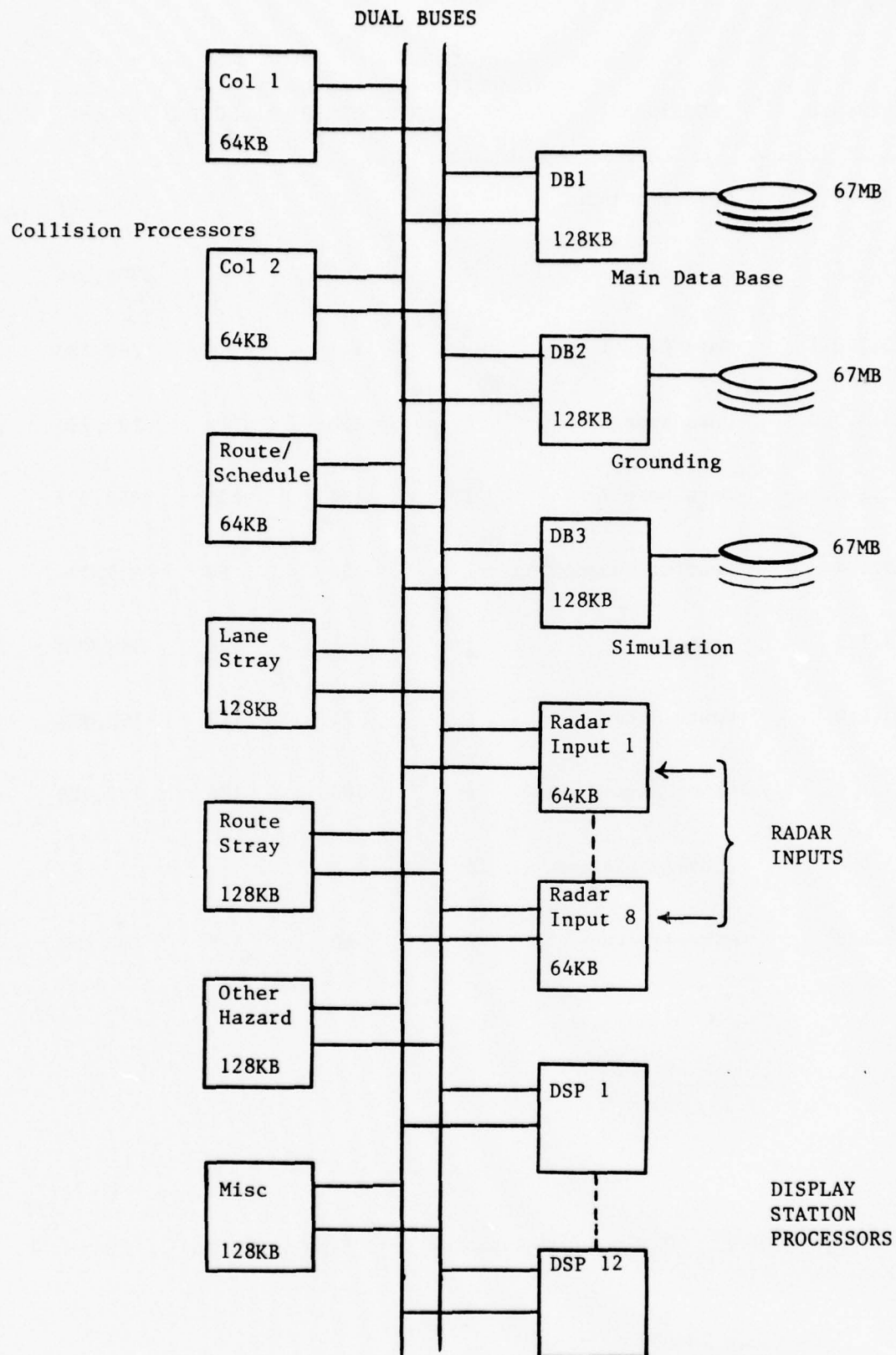


Figure 9-1. Shared Bus, Small Processor Architecture
900 Ships, Class C, Level 4

REFERENCE	PROCESSOR	NUMBER REQUIRED	---- MEMORY ----		CYCLES/ SECOND	CPU UTILIZATION
			REQUIRED K BYTES	ALLOCATED K BYTES		
9.3.1.1	Position Input	8	45	64	486,667	.811
9.3.1.2	Collision	2	39	64	305,524	.509
9.3.1.3.1	Data Base 1	1	111	128	268,280	.447
9.3.1.3.2	Data Base 2	1	108	128	207,780	.346
9.3.1.3.3	Data Base 3	1	124	128	454,097	.757
9.3.1.4	Routing/Scheduling	1	46	64	435,000	.725
9.3.1.5	Lane Stray	1	71	128	360,000	.600
9.3.1.6	Route Stray	1	71	128	360,000	.600
9.3.1.7	Other Hazard/Alert	1	96	128	199,400	.332
9.3.1.8	Display Station	12				
9.3.1.9	Miscellaneous	1	80	128	181,950	.303

Figure 9-2. Processor Summary 900 Ship, Class C, Level 4

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
256 Vessels/6 Seconds @ 10,000 Cycles	426,667	2
Tracker Tables 512 x 32 Bytes		16
CS Nucleus	60,000	16
Bus Interface and Buffers		6
Radar Interface Driver		2
Position Sensor Input Driver		1
System Common		<u>2</u>
TOTAL	<u>486,667</u>	45

Processors with 64k bytes of memory have been selected for the position input processor. The processing load results in a utilization of .811 which is higher than we have attempted to maintain, but is acceptable since the arrival rate of the processing tasks is essentially constant.

9.3.1.2 Collision Processors

Two processors will be required to handle collision processing. The processing load will be evenly divided between the processors by appropriate allocation of the vessels to be considered.

Each processor would maintain position and other data on each of the 900 ships. The collision table would contain an internal ship ID, current system coordinates, x and y components of velocity, and cargo and size codes.

Memory size and processor utilization for the collision processors are given below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Stage 1 - $\frac{1}{2}$ of 485,460	242,730	3
Stage 2 - $\frac{1}{2}$ of 1,800	900	
Stage 3 - $\frac{1}{2}$ of 3,788	1,894	
Collision Table 900 x 12 Bytes		11
OS Nucleus	60,000	16
Bus Interface and Buffers		6
Stage 2 and 3 Processing Queues		1
System Common		<u>2</u>
TOTAL	<u>305,524</u>	<u>39</u>

Processors with 64k bytes of memory have been selected for the Collision processors. Each processor will have a utilization of .509 which provides excess capacity for handling peaks in the number of vessels in Stage 2 and 3 processing.

9.3.1.3 Data Base Processors

These processors will be needed to support functions which require significant data base accessing. All three discs will be kept identical to assure complete backup to the data base. Each data base will receive a copy of all data to be written to disc and will perform all write operations. Disc accesses which require reading will be functionally distributed among the processors.

A portion of the memory requirements and CPU utilization will be common to all three data base processors and is summarized below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Buffer for Grounding Scan, Search, etc.		20
Data Base Buffers		20
OS Nucleus	90,000	16
Bus Interface and Buffers		6
File Handler	17,780	10
Overlay Handler		2
Data Base Manager		10
Disc Driver		2
System Common		<u>2</u>
TOTAL	107,780	88

9.3.1.3.1 Data Base with Grounding and Speed Check

One data base processor will perform the grounding and speed check functions. It will also perform the reads for routing and scheduling.

Processing loads and memory requirements for this processor are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Common Data Base	107,780	88
Grounding and Speed Check	160,500	4
Track Data and Draft 900 x 12 Bytes		11
Basic Cell Data		<u>8</u>
TOTAL	268,280	111

A processor with 128k bytes of memory will be used for data base with grounding. CPU utilization will be approximately .447.

9.3.1.3.2 Main Data Base Processor

The main data base processor will perform normal data base accesses and will support the search on a key function.

Memory requirements and processing loads for the main data base processor are given below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Common Data Base	107,780	88
Search on a Key	100,000	6
Predefined Searches		2
Overlay Area	<u> </u>	<u>12</u>
TOTAL	207,780	108

A processor with 128k bytes of memory will be used for the main data base processor. CPU utilization will be .346.

9.3.1.3.3 Data Base with Simulation

The third data base processor will support simulation and will serve as a backup for either of the other two data bases. It also provides a processor and disc which can be used for program development and other functions which are not normal VTS functions.

Processing loads and memory requirements for the third data base are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Common Data Base	107,780	88
Simulation	346,317	4
Overlay Area		12
Tables for Simulation		<u>20</u>
TOTAL	454,097	124

A processor with 128k bytes of memory will be required. CPU utilization will be .757 when the simulation function is being performed.

9.3.1.4 Routing and Scheduling Processor

One processor will be provided to support routing and scheduling. Data for the routing and scheduling function will be received over the bus from the first data base processor.

Memory requirements and processing loads are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Buffer for Scheduling/Routing		16
Routing/Scheduling	375,000	6
OS Nucleus	60,000	16
Bus Interface and Buffers		6
System Common		<u>2</u>
TOTAL	435,000	46

A processor with 64k bytes of memory will be required. CPU utilization will be .725 which is relatively high. Processing requirements for routing and scheduling should be evaluated more fully during subsequent phases of this study.

9.3.1.5 Lane Stray Processor

One processor will be allocated to perform the Lane Stray function. Memory requirements and processing loads are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Lane Stray	300,000	2
Position Data 900 x 10 Bytes		9
Lane and Route Data		20
Passage Data 2000 x 8 Bytes		16
OS Nucleus	60,000	16
Bus Interface and Buffers		6
System Common		<u>2</u>
TOTAL	<u>360,000</u>	71

A processor with 128k bytes of memory will be provided. CPU utilization will be .600.

9.3.1.6 Route Stray Processor

One processor will be allocated to perform the route stray function. Memory requirements and processing loads are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Route Stray	300,000	2
Position Data 900 x 10 Bytes		9
Lane and Route Data		20
Passage Data 2000 x 8 Bytes		16
OS Nucleus	60,000	16
Bus Interface and Buffers		6
System Common		<u>2</u>
TOTAL	<u>360,000</u>	71

A processor with 128k bytes of memory will be provided. CPU utilization will be .600.

9.3.1.7 Other Hazard and Alert Processor

One processor will be allocated to handle the hazard detection functions which have not been assigned to other processors. This processor will also manage the alert queue and coordinate alert processing with the other hazard detection processors.

Memory requirements and processing loads are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Anchor Drift	4,500	2
Navaid Adrift/Missing	9,900	2
Dangerous Encounters	15,000	2
Excessive Congestion	60,000	2
Alert Response	20,000	4
Alert Queue		4
Watch Circle Data		8
Passage Data 2000 x 8 Bytes		16
Position Data 4000 x 8 Bytes		32
OS Nucleus	90,000	16
Bus Interface and Buffers		6
System Common		2
TOTAL	199,400	96

A processor with 128k bytes of memory will be required for this group of functions. CPU utilization of .332 has been estimated.

9.3.1.8 Display Station Processors

One processor will be required for each of the Display Stations. These processors will support the displays, manage editing for data entry functions and provide an interface with the display station and the remainder of the system. Section 5.3 discusses characteristics of these processors which are assumed to be the same for all the architectures studied. For this case 12 display station processors are assumed.

9.3.1.9 Miscellaneous Processor

One additional processor will be required to support the functions not yet assigned to a processor. Memory sizing and processor loads are summarized below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Encounters	500	2
CPA	300	2
Local Traffic	32,400	3
Environmental Information	5,000	2
Backup Data Update	3,750	2
Vessel Passage History Update	10,000	2
VTs Operations Log	10,000	2
Notices Lookup	10,000	2
Data Entry	20,000	26
Weather Data		1
OS Nucleus	90,000	16
Bus Interface and Buffers		6
History Buffers		12
System Common		<u>2</u>
TOTAL	<u>181,950</u>	<u>80</u>

A processor with 128k bytes of memory will be needed for these functions. A CPU utilization of .303 has been estimated.

9.3.2 Interprocessor Communication Loads

An estimate of the interprocessor communications loads has been made to provide an approximation of the bandwidth required for the bus or other shared link. These loads are summarized in Figure 9-3 and discussed below.

Position Input Data

Position input data for all vessels and buoys which are in track will be transmitted from the position input processors to the hazard/alert processor. In determining the communications load we have assumed that complete tracker input tables will be transmitted every position update cycle (6 seconds). The load then will be the product of: 32 bytes per entry; 512 entries per radar; and 8 radars, for a total load of 131,072 bytes every 6 seconds.

Vessel Position Data

Vessel position data is a subset of the total position input data which will be distributed by the hazard/alert processor to the other processors which need this data. The 2 collision processors, the lane stray and route stray processors, and the data base processors will receive copies of the vessel position data. An additional copy of the vessel position data will be distributed to the appropriate display stations.

Approximately 10 bytes of data for each of 900 ships will be distributed to 8 different locations for a total load of 72,000 bytes every six seconds.

	<u>BYTES</u>	<u>TIME SECONDS</u>	<u>BYTES PER SECOND</u>	<u>BITS PER SECOND</u>
Position Input Data	131,072	6	21,845	174,760
Vessel Position Data	72,000	6	12,000	96,000
Data Base Writes	2,928	1	2,928	23,424
Watchstander Requests	300	1	300	2,400
Display Replace	60,000	6	10,000	80,000
Scheduling and Routing	84,450	1	84,450	675,600
Alert Notices	1,000	1	1,000	8,000
Status Messages	1,000	1	1,000	8,000
				<hr/> 1,068,184
Overhead - 50%				534,092
				<hr/>
TOTAL				1,602,276

Figure 9-3. Interprocessor Communications Load

Data Base Writes

Copies of the data to be written to the disc must be sent from the main data base processor to the secondary data base processors. Assuming 1.43 writes per second (Figure 4-11) at 1024 bytes per write this gives a load of 1464 bytes per second per data base or 2928 bytes per second.

Watchstander Requests

Approximately 300 bytes per second can be assumed as the peak for messages from the watchstanders to the main data base processor.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 30,000 bytes of data must be transmitted within a six second time period for each display replaced.

The computed total load assumes that two display replacements can occur at any given time. This assumption is reasonable since the frequency of display replacements is expected to be low. It should be noted, however, that the bandwidth required of the processor interconnection would be increased significantly if provision is made to allow more than two displays to be replaced simultaneously.

Scheduling and Routing

Scheduling and routing will add a significant interprocessor communication load. Data from the data base will be retrieved from the disc and transmitted to the scheduling and routing processor. Peak loadings are based on 5.63 disc accesses per second at 15,000 bytes per access for a total of 84,450 bytes per second.

Alert Notices

Information must be transmitted from the hazard detecting processors to the alert processor. A rate of 1000 bytes per second has been assumed.

Status Messages

Status messages will be sent periodically from processor to processor throughout the system. These messages will be used to determine the loading and current health of the processors. A rate of 1000 bytes per second should be sufficient for this function.

Overhead

Message source and destination information, checksums and other overhead will be added to the interprocessor messages handled by the system. It may also be desirable to send positive acknowledgements to each message. An overhead rate of 50% should be adequate.

9.3.3 Disc Sizing and Utilization

Moving head discs will serve as the mass storage devices for the data base discussed in Section 4. All discs will be assumed to be of the same type and sized to handle the maximum case. Discs of the 3330 type, which are available from a number of manufacturers, have been assumed. The exact characteristics of the discs vary from manufacturer to manufacturer. The following characteristics are typical:

- . 67 megabytes of storage
- . 16.67 ms rotation time
- . 30 ms average to locate a cylinder
- . 20480 bytes per track

9.3.3.1 Disc Utilization Factors

In this section we will develop estimates of the disc utilization resulting from a number of system functions. The estimates are summarized in Figure 9-4.

Utilization for each of the functions can be determined by multiplying the service time per access by the number of accesses per unit time.

Service time for an access, T_s , is the sum of the time to locate the cylinder, the time to locate the data on the cylinder (which averages one half a revolution) and the time to transfer the data. Numerically this is

$$T_s = 30 \text{ ms} + \frac{16.67 \text{ ms}}{2} + \frac{\text{Bytes}}{20480 \text{ Bytes/Rev}} \times 16.67 \text{ ms/rev}$$

which reduces to:

$$T_s = 38.33 + \frac{\text{Bytes}}{1228.6} \text{ (ms)}.$$

9.3.3.1.1 Grounding

For grounding we assume that the entire cell data base is read every 30 seconds in units of 20,480 bytes, or one track. Service time would average 55 milliseconds per access which yields a utilization of .229 assuming 4.17 accesses per second.

9.3.3.1.2 Scheduling and Routing

Execution of the Scheduling and Routing function results in a data base access rate of 5.63 accesses per second. Each access involves reading approximately 15,000 bytes. These records are read to allow a proposed route and schedule to be compared against all previously determined routes and schedules. Utilization will be .284 based on a computed service time of 50.5 milliseconds.

	<u>TYPICAL RECORD SIZE BYTES</u>	<u>SERVICE TIME MS</u>	<u>ACCESSES PER SECOND</u>	<u>UTILIZATION</u>
Grounding	20,480	55.0	4.17	.229
Scheduling/Routing	15,000	50.5	5.63	.284
Data Base Search	5,120	42.5	4.17	.177
Normal Reads	1,024	39.2	6.26	.245
Normal Writes	1,024	39.2	1.43	.056

Figure 9-4. Disc Utilization Factors

9.3.3.1.3 Data Base Search

The search on a key function will be performed by reading the entire file and comparing selected fields against the limits and/or ranges specified by the watchstander.

The selected file will be read in blocks of records and will require an average of 4.17 accesses per second of 5120 bytes each for the vessel file, which is the worst case. Utilization will be .177 for the computed service time of 42.5 milliseconds.

9.3.3.1.4 Normal Reads

Normal reads include support for all the standard watchstander functions such as enter vessel or delete passage. From Section 4, the estimated design rate will be 6.26 accesses per second. Since the most common access is reading an index block of 1024 bytes the utilization will be .245 for the computed service time of 39.2 milliseconds.

9.3.3.1.5 Normal Writes

Normal writes include all the functions which alter the data base. These functions will be duplicated by each of the data base processors to allow immediate switchover to an alternate in case of disc or processor failure.

From the rate determined in Section 4 of 1.43 writes/second and assuming an average of 1024 bytes per operation a utilization of .056 is computed for a service time of 39.2 milliseconds.

9.3.3.2 Disc Functional Allocation

The utilization factors developed in the preceding section would give a total utilization of .991 if all functions accessed the same disc. Such a utilization would result in extraordinarily long response times.

To reduce utilization and make response times acceptable, two discs will be required to handle the load with an additional disc to serve essentially as a spare.

The first disc would handle grounding and routing and scheduling as well as writing all altered blocks to keep it identical to the main data base. Total utilization of this disc would then be .569.

The second disc would handle all normal reads and writes and would support the data base searches. Total utilization would be .478.

The third disc would be kept current by performing all write operations and would support simulation, program development, and other non-critical functions while serving as an on-line backup to the other two discs.

9.4 CLASS B, LEVEL 4 SYSTEM

In this section a preliminary design is presented for a Class B, Level 4 system capable of handling 150 identified and 350 unidentified vessels. The structure of this system is shown in Figure 9-5.

The system is similar to the one described for a Class C, Level 4 system. The number of processors is reduced, however, because of a reduction in the processing load and memory requirements which results from the lesser number of ships and from eliminating the routing and scheduling functions.

9.4.1 Processor Sizing and Utilization

Processors have been assigned functions in the same manner as for the Class C system. In some cases, however, the lesser number of ships allows us to combine functions performed by two or more processors in the Class C system. Memory sizes and CPU utilizations for each processor have been summarized in Figure 9-6.

9.4.1.1 Position Input Processor

Position input processors will be identical to those specified for the Class C system (see Section 9.3.1.1). Three processors supporting three radars should be sufficient to handle the 150 identified and 350 unidentified vessels.

9.4.1.2 Data Base Processors

Two processors will be needed to support functions which require significant data base accessing.

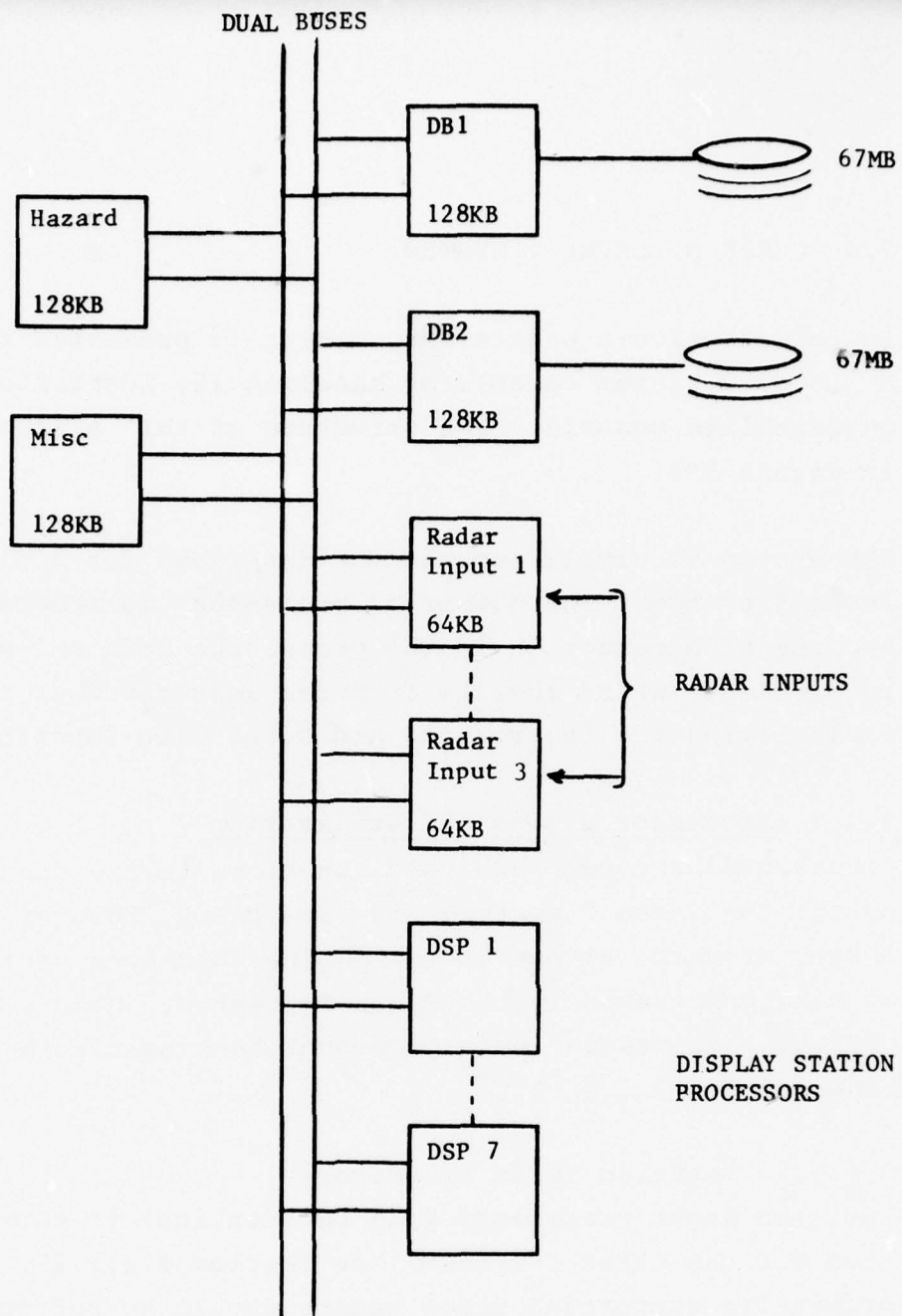


Figure 9-5. Shared Bus, Small Processor Architecture
500 Ships, Class B, Level 4

REFERENCE	PROCESSOR	NUMBER REQUIRED	---- MEMORY ---- REQUIRED K BYTES	ALLOCATED K BYTES	CYCLES/ SECOND	CPU UTILIZATION
9.4.1.1	Position Input	3	45	65	486,667	.811
9.4.1.2.1	Data Base 1	1	118	128	219,444	.366
9.4.1.2.2	Data Base 2	1	94	128	438,861	.731
9.4.1.3	Hazard And Alert	1	92	128	312,648	.521
9.4.1.4	Display Station	7				
9.4.1.5	Miscellaneous	1	80	128	134,950	.225

Figure 9-6. Processor Summary
500 Ships, Class B, Level 4

As previously described (Section 9.3.1.3) the two data bases will be identical to assure rapid recovery from a failure.

A portion of the memory requirements and CPU utilization will be common to both data base processors and is summarized below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Data Base Buffers		10
OS Nucleus	90,000	16
Bus Interface and Buffers		6
File Handler	2,544	10
Overlay Handler		2
Data Base Manager		10
Disc Driver		2
System Common		2
Overlay Area		<u>12</u>
TOTAL	<u>92,544</u>	70

9.4.1.2.1 Main Data Base

One data base processor will serve as the main data base processor combining the main data base (Section 9.3.1.3.2) and the data base with grounding and speed check (Section 9.3.1.3.1) of the Class C system. Processing loads and memory requirements for this processor are shown below.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Common Data Base	92,544	70
Buffer for Grounding		20
Search Buffer		.6
Grounding and Speed Check	26,900	4
Tracker Data and Draft 150 x 12 bytes		2
Basic Cell Data		8
Search on a Key	100,000	6
Predefined Searches		2
TOTAL	219,444	118

A processor with 128k bytes of memory will be required. CPU utilization is estimated at .366.

9.4.1.2.2 Data Base with Simulation

The second data base processor will support simulation and will serve as an online backup to the main data base processor. It is essentially the same as the corresponding processor (Section 9.3.1.3.3) in the Level C system.

Estimated memory requirements and processing loads, which are shown below, are slightly less than for the Class C version.

	<u>Cycles/Second</u>	<u>Memory</u> <u>K Bytes</u>
Common Data Base	92,544	70
Simulation	346,317	4
Simulation Tables		20
TOTAL	438,861	94

A processor with 128k bytes of memory will be used. CPU utilization will be .731.

9.4.1.3 Hazard and Alert Processor

One processor will handle all hazard and alert processing except the grounding and speed limit checks which are performed by the main data base processor. It combines the functions of several processors in the 900 ship Level C system including: Collision processors (Section 9.3.1.2); Lane stray processor (Section 9.3.1.5); Route stray processor (Section 9.3.1.6); and the other hazard and alert processor (Section 9.3.1.7).

Memory size and processing loads for the hazard and alert processor are shown below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Collision Stage 1	76,428	3
Stage 2	900	
Stage 3	2,020	
Collision Table 150 x 12 bytes		2
OS Nucleus	90,000	16
Bus Interface and Buffers		6
Stage 2 and 3 Processing Queue		1
System Common		2
Lane Stray	50,000	2
Route Stray	50,000	2
Passage Data 1000 x 8 bytes		8
Lane and Route Data		20
Anchor Drift	900	2
Navaid Adrift/Missing	9,900	2
Dangerous Encounters	2,500	2
Excessive Congestion	10,000	2
Alert Response	20,000	4
Watch Circle Data		2
Alert Queue		4
Position Data 8 x 1500		<u>12</u>
TOTAL	<u>312,648</u>	92

One processor with 128k bytes will be used for hazard and alert processing. CPU utilization for this processor is estimated to be .521.

9.4.1.4 Display Station Processor

Display station processors will be identical with those discussed in Section 9.3.1.8. Seven processors are assumed including a watch supervisor and a spare station.

9.4.1.5 Miscellaneous Processor

The miscellaneous processor is functionally identical to the one described in Section 9.3.1.9. The reduced number of vessels supported reduces the CPU utilization to .225 with the allocated memory remaining at 128k bytes.

9.4.2 Interprocessor Communication Loads

An estimate of the interprocessor communications loads has been made to provide an approximation of the bandwidth required for the bus or other shared link. These loads are summarized in Figure 9-7 and discussed below.

Position Input Data

Position input data for all vessels and buoys which are in track will be transmitted from the position input processors to the hazard/alert processor. In determining the communications load we have assumed that complete tracker input tables will be transmitted every position update (6 seconds). The load will be the product of: 32 bytes per entry; 512 entries per radar; and 3 radars for a total load of 49,152 bytes every 6 seconds.

	<u>BYTES</u>	<u>TIME SECOND</u>	<u>BYTES PER SECOND</u>	<u>BITS PER SECOND</u>
Position Input Data	49,152	6	8,192	65,536
Vessel Position Data	15,000	6	2,500	20,000
Data Base Writes	246	1	246	1,968
Watchstander Requests	150	1	150	1,200
Display Replace	60,000	6	10,000	80,000
Status Messages	500	1	500	4,000
				<hr/> 172,704
Overhead - 50%				86,352
TOTAL				<hr/> 259,056

Figure 9-7. Interprocessor Communications Load
500 Ships, Class B, Level 4

Vessel Position Data

Vessel position data is a subset of the total position input data which will be distributed by the hazard/alert processor to the other processors which need this data. The data base processors and the appropriate display stations will receive copies.

Approximately 10 bytes of data for each of 500 ships will be distributed to 3 different locations for a total load of 15,000 bytes every 6 seconds.

Data Base Writes

Copies of the data to be written to the disc must be sent from the main data base processor to the secondary data base processor. Assuming .24 writes per second (Figure 4-12) at 1024 bytes per write this gives a load of 246 bytes per second.

Watchstander Requests

Approximately 150 bytes per second will result from messages from the watchstander to the main data base processor.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 60,000 bytes of data must be transmitted within a six second time period.

Status Messages

Status messages will be sent periodically from processor to processor throughout the system. These messages will be used to determine the loading and current health of the processor. A rate of 500 bytes per second should be sufficient for this function.

Overhead

Message source and destination information, checksums and other overhead will be added to the interprocessor messages handled by the system. Positive acknowledgements may also be desirable. An overhead rate of 50% should be adequate.

9.4.3 Disc Sizing and Utilization

Moving head discs will serve as the mass storage devices for the data base discussed in Section 4. The discs will be identical to the discs used in the Class C, Level 4 system (see Section 9.3.3). Actual disc storage requirements will be somewhat less than required for the Level C systems. The cost savings which might be achieved by using smaller discs is insufficient to justify using different discs.

Disc utilization for the Class B system is shown in Figure 9-8. (See Section 9.3.3.1 for calculation method). Total utilization is estimated to be .456 which indicates that a single disc will be sufficient to support data base accessing.

	TYPICAL RECORD SIZE BYTES	SERVICE TIME MS	ACCESSES PER SECOND	UTILIZATION
Grounding	20,480	55.0	4.17	.229
Data Base Search	5,120	42.5	4.17	.177
Normal Reads	1,024	39.2	1.04	.041
Normal Writes	1,024	39.2	.24	.009
TOTAL				.456

Figure 9-8. Disc Utilization Factors
500 Ships, Class B, Level 4

9.5 CLASS A, LEVEL 1 SYSTEM

In this section a preliminary design is presented for a Class A, Level 1 system capable of handling 100 vessels. The structure of this system is shown in Figure 9-9.

The system is similar to the ones described for Class C, Level 4 and Class B, Level 4. The number of processors is further reduced because of the lesser number of ships and the elimination of the functions associated with radar and other automated position sensors.

9.5.1 Processor Sizing and Utilization

Five display stations have been assumed for the Class A, Level 1 system supporting 100 vessels. The display station processors will be identical to those discussed in Section 9.3.1.8. Three additional processors are required. Memory sizing and CPU utilization are discussed below and summarized in Figure 9-10.

9.5.1.1 Main Processors

Two main processors, each capable of supporting the data base, will be provided. One of the processors will handle the data base and a number of the other system functions. The second main processor will maintain its copy of the data base and serve as a hot standby capable of taking over the functions of the other main processor or the miscellaneous processor in the event of a failure of either.

DUAL BUSES

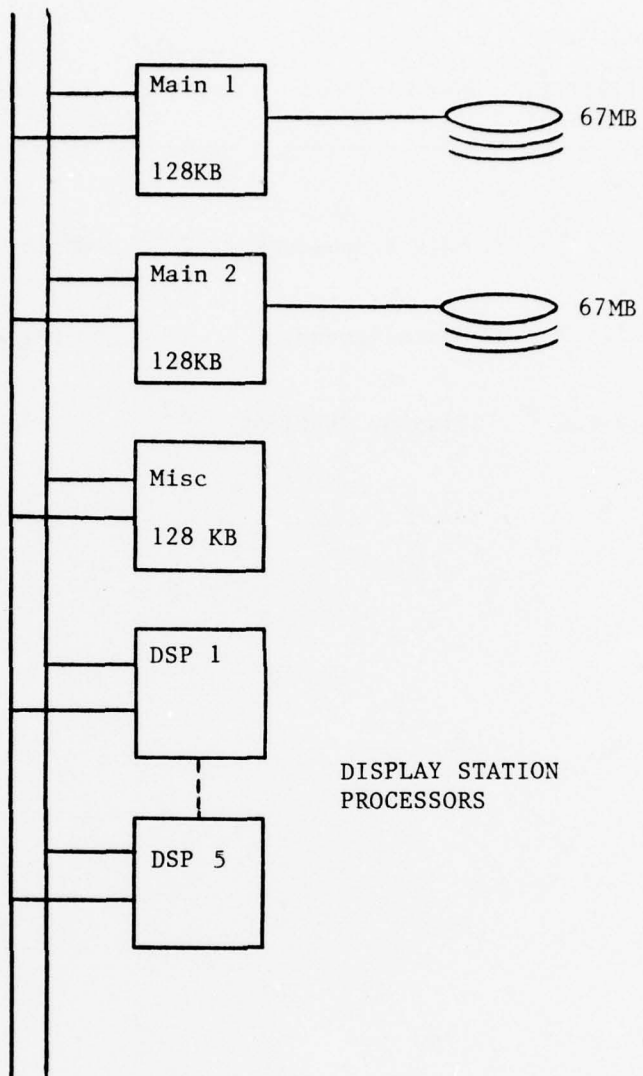


Figure 9-9. Shared Bus, Small Processor Architecture
100 Ships, Class A, Level 1

REFERENCE	PROCESSOR	NUMBER REQUIRED	---- MEMORY ---- REQUIRED K BYTES	ALLOCATED K BYTES	CYCLES/ SECOND	CPU UTILIZATION
9.5.1.1	Main Processors	2	124	128	291,660	.486
9.5.1.2	Miscellaneous	1	102	128	123,554	.207
9.3.1.8	Display Stations	5				

Figure 9-10. Processor Summary
100 Ships, Class A, Level 1

Memory sizing and processing loads for the main processor are summarized below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Search Buffer		6
Data Base Buffers		8
OS Nucleus	90,000	16
Bus Interface and Buffers		6
File Handler	1,540	10
Overlay Handler		2
Overlay Area		12
Data Base Manager		10
Disc Driver		2
System Common		2
Search on a Key	100,000	6
Predefined Searches		2
Simulation	79,620	4
Simulation Tables		20
Backup Data Update	500	2
Passage History Update	10,000	2
VTS Operations Log	10,000	2
History Buffers		<u>12</u>
TOTAL	291,660	124

The main processor will have 128k bytes of memory. The estimated CPU utilization is .486.

9.5.1.2 Miscellaneous

Because of memory limitations an additional processor will be assigned to handle the functions which could not be supported

by the main processors. It should be noted that this processor is not functionally equivalent to the miscellaneous processor in the Class C and Class B systems.

Memory sizing and processing loads are summarized below.

	<u>Cycles/Second</u>	<u>Memory K Bytes</u>
Encounters	9	2
Local Traffic	3,600	3
Notices Lookup	170	2
Environmental Information	85	2
Position Update	360	2
OS Nucleus	90,000	16
Bus Interface and Buffers		6
System Common		2
Route Stray	360	2
Passage Data		6
Lane and Route Data		20
Dangerous Encounter	1,500	2
Excessive Congestion	6,000	2
Alert Response	20,000	4
Alert Queue		4
Data Entry	2,000	26
Weather Data		<u>1</u>
TOTAL	<u>124,084</u>	102

A processor with 128k bytes of memory will be needed. CPU utilization will be .207.

9.5.2 Interprocessor Communications Load

The interprocessor communications loads will be less than the load calculated for the Class B, Level 4 system (Section 9.4.2).

The load generated when a display is replaced will predominate. A bandwidth of approximately 200,000 bits per second will be required.

9.5.3 Disc Utilization and Sizing

Discs will be assumed to be of the same type and size previously specified for the Class B and Class C systems (See Sections 9.3.3 and 9.4.3). Disc utilization is summarized in Figure 9-11.

9.6 RELATIVE HARDWARE COSTS

Figure 9-12 summarizes the relative hardware costs for the three cases of Architecture I. Costs for processors, memory, disc drives and interprocessor connections are included. Costs for hardware which is common to all architectures are not included. The costs of the display processors and display hardware is specifically excluded. The cost for interconnection of the display stations processors is included.

The method used for estimating unit costs for hardware components is explained in Appendix 1.

	<u>TYPICAL RECORD SIZE BYTES</u>	<u>SERVICE TIME MS</u>	<u>ACCESSES PER SECOND</u>	<u>UTILIZATION</u>
Data Base Search	5,120	42.5	4.17	.177
Normal Reads	1,024	39.2	.74	.029
Normal Writes	1,024	39.2	.16	.006
				<hr/>
TOTAL				.212

Figure 9-11. Disc Utilization Factors
100 Ships, Class A, Level 1

<u>Class C, Level 4</u>	<u>Number</u>	<u>Unit Cost</u> (<u>\$</u>)	<u>Total</u> (<u>\$</u>)
<u>Processors</u>			
64KB	11	11,000	121,000
128KB	7	18,000	126,000
<u>Discs</u>			
67MB	3	25,000	75,000
<u>Bus Couplers</u>	60	3,000	<u>180,000</u>
TOTAL			502,000
<u>Class B, Level 4</u>			
<u>Processors</u>			
64KB	3	11,000	33,000
128KB	4	18,000	72,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Bus Couplers</u>	28	3,000	<u>84,000</u>
TOTAL			239,000
<u>Class A, Level 1</u>			
<u>Processors</u>			
128KB	3	18,000	54,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Bus Couplers</u>	16	3,000	<u>48,000</u>
TOTAL			152,000

Figure 9-12. Cost Summary: Shared Bus, Small Processor Architecture

The second candidate architecture we will consider uses more powerful processors than were used in the first architecture. The processors are interconnected by a high speed shared link. System functions are distributed among the processors to form a distributed function multiple processor.

10.1 PHYSICAL STRUCTURE

The physical structure of the second candidate architecture is described below in terms of the processors and the interconnection mechanism.

10.1.1 The Processors

Relatively high powered 16 bit minicomputers were selected for this architecture. Processors of this type can handle approximately 1,250,000 instruction cycles per second. With memory mapping, up to a million bytes of main memory can be supported by a single processor. Floating point hardware is normally available and has been assumed for the two larger cases studied.

Processors with the above characteristics are available from a wide variety of manufacturers.

Thirty-two bit minicomputers are also available from a number of manufacturers. Thirty-two bit minis provide roughly two to four times the processing power of the 16 bit minis with the relative processing power a function of the complexity of the functions performed. The increased address space eliminates the software complexity added by memory mapping. It seems probable that the increased cost of the thirty-two bit minis can not be justified for the typical VTS. Further study of the tradeoffs

between 16 and 32 bit minis may be made at a later time if either architecture II or architecture III is selected. For the purposes of this architecture study, 16 bit minis will be assumed.

10.1.2 Interconnection

Processors will be interconnected by a shared link which will be duplicated to assure availability. The shared link could be an 8 or 16 bit bus or a shared serial line. The required bandwidth for the shared link will be discussed in Section 10.3.2.

10.2 LOGICAL STRUCTURE

Logically the system consists of a number of processors each performing a fixed group of functions.

In order to minimize the interdependence of processors and reduce interprocessor communications, functions that use the same devices or data are grouped together.

Exception functions are allocated to the OS nucleus and the relative position function is allocated to the DSP, as for Architecture I.

10.3 CLASS C, LEVEL 4 SYSTEM

In this section a preliminary design is presented for a Class C, Level 4 system capable of handling 900 identified vessels. The structure of the system is shown in Figure 10-1.

10.3.1 Processor Sizing and Utilization

Functions have been assigned to processors in related functional groups. Four functional groups will be needed for the Class C, Level 4 system.

CPU utilizations will be kept below 70%* to prevent difficulties in meeting response time requirements.

The following sections describe the processors which will be required for the 900 ship, Class C, Level 4 system. Memory sizes and CPU utilizations have been estimated for each of the processors and are summarized in Figure 10-2.

10.3.1.1 Position/Collision Processor

The position/collision processor performs two of the primary system functions: Position input processing and collision detection. All eight radars will be interfaced to the position/collision processor and to the backup processor. The position/collision processor will handle coordinate transformation and other radar support functions. All collision processing will also be performed by the position/collision processor.

*For the first architecture, using the less powerful processors, we attempted to keep CPU utilizations near 50%. The faster processors can give the same response times at a higher utilization because average service time is reduced.

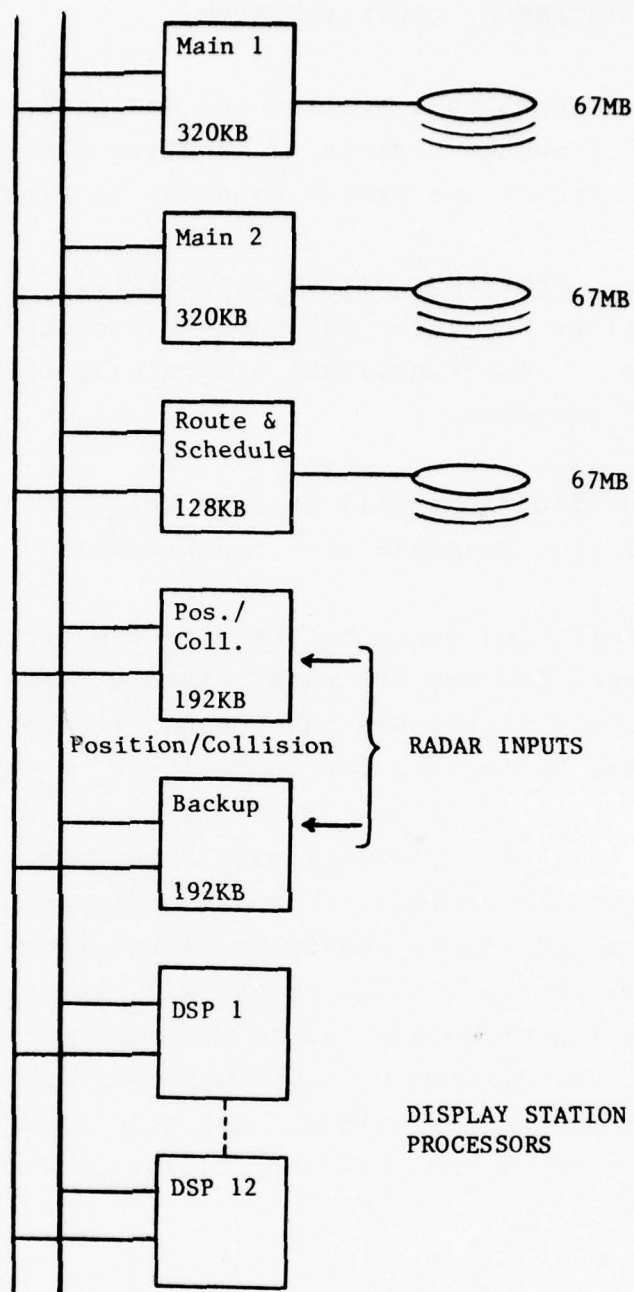


Figure 10-1. Shared Bus, Large Mini Architecture
900 Ships, Class C, Level 4

REFERENCE	PROCESSOR	NUMBER REQUIRED	---- MEMORY ---- REQUIRED K BYTES	ALLOCATED K BYTES	CYCLES/ SECOND	CPU UTILIZATION
10.3.1.1	Position/ Collision	2	173	192	828,548	.663
10.3.1.2	Display Stations	12				
10.3.1.3.1	Scheduling/ Routing	1	104	128	580,280	.464
10.3.1.3.2	Main	2	278	320	783,807	.627

Figure 10-2. Processor Summary
900 Ships, Class C, Level 4

Memory sizing and processor utilization for the position/collision processor are tabulated below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Collision Stage 1	485,460	3
Stage 2	1,800	
Stage 3	3,788	
Position Update	150,000	2
Tracker Tables 8 x 512 x 32 Bytes		128
Collision Table 900 x 12 Bytes		11
OS Nucleus	187,500	16
Bus Interface and Buffers		6
Radar Interface Driver		2
Position Sensor Input Driver		1
System Common		2
Memory Mapping		2
TOTAL	828,548	173

A processor with 192K bytes of memory will be required to support the position input and collision detection functions. CPU utilization is estimated to be .663.

10.3.1.2 Display Station Processors

As in the other architectures being considered, one processor has been assumed for each of the Display Stations. These processors will support the displays, manage editing for data entry functions and provide an interface between the display station and the remainder of the system. A preliminary look at the characteristic of the display station processors is given in Section 5.3. Since this group of processors is assumed to be identical for all architectures studied, the precise characteristics are not significant in the evaluation of alternative architectures.

10.3.1.3 Processors with Data Base

The remaining processors will support moving head discs and will be capable of supporting data base operations. The following memory and processing will be common to these processors.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
OS Nucleus	187,500	16
Bus Interface and Buffers		6
System Common		2
Data Base Buffers		20
File Handler	17,780	10
Overlay Handler		2
Overlay Area		12
Data Base Manager		10
Disc Drivers		2
Memory Mapping		<u>2</u>
TOTAL	<u>205,280</u>	<u>82</u>

10.3.1.3.1 Scheduling and Routing Processor

One processor will provide the processing and data base facility for routing and scheduling. In addition, the scheduling and routing processor will maintain a current copy of the data base as a backup in case of a failure of one of the other two processors. Memory sizing and processing load is summarized below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Data Base Common	205,280	82
Buffer for Scheduling		16
Scheduling and Routing	<u>375,000</u>	<u>6</u>
TOTAL	<u>580,280</u>	<u>104</u>

A processor with 128K bytes of memory will be required. CPU utilization will be about .464.

10.3.1.3.2 Main Processor

Two main processors will be provided. Each will be capable of performing all the remaining functions of the system. Because of high disc utilization (see Section 9.3.3.1) the grounding function will be assigned to one of the processors while the other is assigned the data base searches. Both will maintain a complete current copy of the data base to allow a reallocation of functions in the event of a failure.

Processing loads and memory size estimates are given below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Data Base Common	205,280	82
Lane Stray	30,000	2
Route Stray	30,000	2
Dangerous Encounters	15,000	2
Excessive Congestion	6,000	2
Anchor Drift	4,500	2
Navaid Adrift/Missing	9,900	2
Encounters	500	2
CPA	300	2
Local Traffic	32,400	3
Environmental Information	5,000	2
Alert Responses	20,000	4
Simulation	130,677	4
Simulation Tables		20
Grounding and Speed Limit Checks	160,500	4
System Backup Data	3,750	2
Vessel Passage History	10,000	2
VTs Operations Log	10,000	2

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Notices Lookup	10,000	2
Search on a Key	100,000	6
Predefined Searches		2
Grounding Buffer		20
Search Buffer		6
Position Data 4000 x 8 bytes		32
Lane and Route Data		20
Basic Cell Data		8
Passage Data 2000 x 8 bytes		16
Watch Circle Data		8
Alert Queue		4
Weather Data		1
History Buffers		<u>12</u>
TOTAL	<u>783,807</u>	<u>278</u>

Processors with 320K bytes of memory will be required. CPU utilization will be .627 when all processing is performed by one processor.

10.3.2 Interprocessor Communication Loads

An estimate of the interprocessor communications load has been made to provide an approximation of the bandwidth required for the bus or other shared link. These loads are summarized in Figure 10-3 and discussed below.

Position Input Data

Position input data for all vessels and buoys which are in track will be transmitted from the position/collision processor to the main processor. In determining the communications load we have assumed that complete tracker input tables will be transmitted every position update cycle (6 seconds). The load then will be the product of: 32 bytes per entry; 512 entries per radar; and 8 radars for a total load of 131,072 bytes every 6 seconds.

Data Base Writes

Copies of the data to be written to the disc must be sent from the main data base processor to the secondary data base processors. Assuming 1.43 writes per second (Figure 4-11) at 1024 bytes per write this gives a load of 1464 bytes per second per data base or 2928 bytes per second.

Watchstander Requests

Approximately 300 bytes per second can be assumed as the peak for messages from the watchstanders to the main data base processor.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 30,000 bytes of data must be transmitted within a six second time period for each map altered.

The computed total load assumes that two display replacements can occur at any given time. This assumption is reasonable since the frequency of display replacements is expected to be low. It should be noted, however, that the bandwidth required of the processor interconnection would be increased significantly if provision is made to allow more than two displays to be replaced simultaneously.

Alert Notices

Information must be transmitted from the hazard detecting processor to the alert processor. A rate of 1000 bytes per second has been assumed.

	<u>Bytes</u>	<u>Time Seconds</u>	<u>Bytes per Second</u>	<u>Bits per Second</u>
Position Input Data	131,072	6	21,845	174,760
Data Base Writes	2,928	1	2,928	23,424
Watchstander Requests	300	1	300	2,400
Display Replace	60,000	6	10,000	80,000
Alert Notices	1,000	1	1,000	8,000
Status Messages	1,000	1	1,000	<u>8,000</u>
				296,584
Overhead - 50%				<u>148,292</u>
TOTAL				<u>444,876</u>

Figure 10-3. Interprocessor Communications Load

Status Messages

Status messages will be sent periodically from processor to processor throughout the system. These messages will be used to determine the loading and current health of the processors. A rate of 1000 bytes per second should be sufficient for this function.

Overhead

Message source and destination information, checksums and other overhead will be added to the interprocessor messages handled by the system. It may also be desirable to send positive acknowledgements to each message. An overhead rate of 50% should be adequate.

10.3.3 Disc Sizing and Utilization

Moving head discs selected for this architecture will be identical to those selected for Candidate Architecture I (See Section 9.3.3).

Disc utilization factors will be the same as those calculated in Section 9.3.3.1 and summarized in Figure 9-4. Assignment of functions to the individual discs will be somewhat different from the assignments for Architecture I (See Section 9.3.3.2).

The disc associated with the first main processor will handle data base searches and all normal reads and writes for a total utilization of .478.

The disc associated with second main processor will support the grounding functions and maintain a current copy of the data base. Total utilization will be .285.

The third disc will be associated with the routing and scheduling processor. It will also maintain a current copy of the data base. Total utilization will be .340.

10.4 CLASS B, LEVEL 4 SYSTEM

In this section a preliminary design is presented for a Class B, Level 4 system capable of handling 150 identified and 350 unidentified vessels. The structure of this system is shown in Figure 10-4.

The system is similar to the one described for the Class C, Level 4 system except that the number of processors is reduced.

10.4.1 Processor Sizing and Utilization

Two functional groups of processors will be required for the Class B, Level 4 system.

10.4.1.1 Display Station Processor

Display Station processors are required as for all other classes of systems and for all the architectures under consideration. Seven display stations are assumed for the class B, Level 4 system including a watch supervisor and a training station.

10.4.1.2 Main Processor

Two main processors will be required for the Class B, Level 4 system. Each will be capable of handling the total processing load. One of the processors will serve as an on-line backup with the other processor handling the total processing load.

Processor utilization and memory requirements are summarized below.

DUAL BUSES

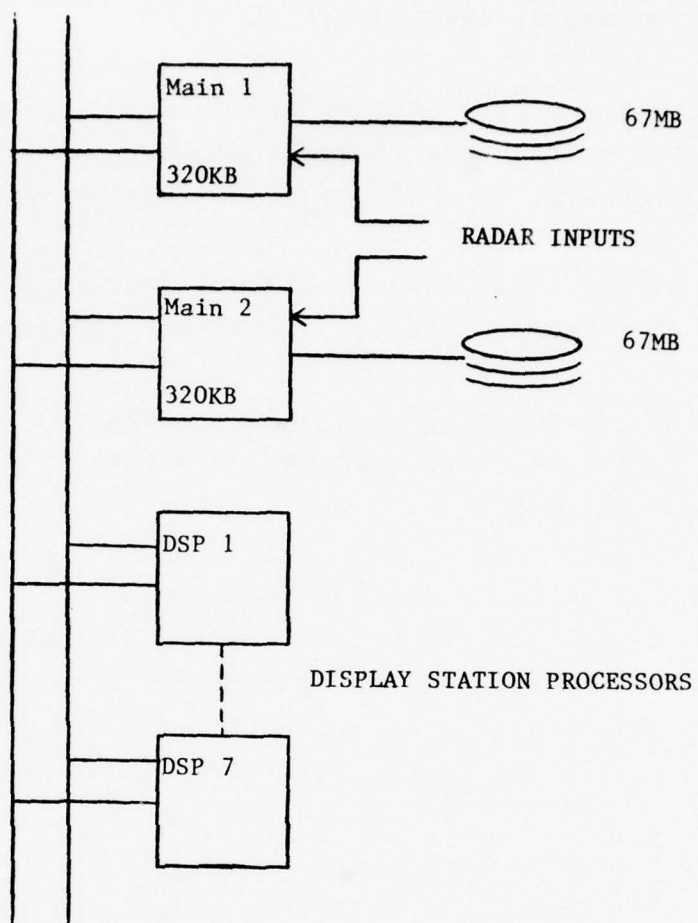


Figure 10-4. Shared Bus, Large Mini Architecture
500 Ships, Class B, Level 4

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Collision Stage 1	76,428	3
Stage 2	900	
Stage 3	2,020	
Position Update	83,000	2
Tracker Tables 3 x 512 x 32 Bytes		48
Collision Table 500 x 12 Bytes		6
Radar Interface Driver		2
Position Sensor Input Driver		1
OS Nucleus	187,500	16
Bus Interface and Buffers		6
System Common		2
Data Base Buffers		10
File Handler	2,544	10
Overlay Handlers		2
Overlay Area		12
Data Base Manager		10
Disc Driver		2
Memory Mapping		2
Lane Stray	5,000	2
Route Stray	5,000	2
Dangerous Encounters	2,500	2
Excessive Congestion	1,000	2
Anchor Drift	900	2
Navaid Adrift/Missing	9,900	2
Encounters	500	2
CPA	300	2
Local Traffic	5,400	3
Environmental Information	5,000	2
Alert Responses	20,000	4
Simulation	130,677	4

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Simulation Tables	26,900	20
Grounding and Speed Limit Check		4
System Backup Data	750	2
Vessel Passage History	10,000	2
VTs Operations Log	10,000	2
Notices Lookup	10,000	2
Search on a Key	100,000	6
Search Buffer		6
Predefined Searches		2
Grounding Buffer		20
Position Data 1500 x 8 Bytes		12
Lane and Route Data		20
Basic Cell Data		8
Passage Data 1000 x 8 Bytes		8
Watch Circle Data		2
Alert Queue		4
Weather Data		1
History Buffers		<u>12</u>
TOTAL	<u>696,219</u>	<u>296</u>

Processors with 320K bytes of memory will be needed. CPU utilization for the processor handling the load will be .557.

10.4.2 Interprocessor Communication Loads

An estimate of the interprocessor communications loads has been made to provide an approximation of the bandwidth required for the bus or other shared link. These loads are summarized in Figure 10-5 and discussed below.

	<u>Bytes</u>	<u>Time Seconds</u>	<u>Bytes Per Second</u>	<u>Bits Per Second</u>
Vessel Position Data	5,000	6	833	6,664
Data Base Writes	246	1	246	1,968
Watchstander Request	150	1	150	1,200
Display Replace	60,000	6	10,000	80,000
Status			500	<u>4,000</u>
				93,832
Overhead - 50%				<u>46,916</u>
TOTAL				140,748

Figure 10-5. Interprocessor Communications Load
500 Ship, Class B, Level 4

Vessel Position Data

Updated vessel position data will be transmitted from the main processor to the appropriate display stations. Approximately 10 bytes of data for each of 500 ships will be distributed to the display stations for a total load of 5,000 bytes every 6 seconds.

Data Base Writes

Copies of the data to be written to the disc must be sent from the main data base processor to the secondary data base processor. Assuming .24 writes per second (Figure 4-12) at 1024 bytes per write this gives a load of 246 bytes per second.

Watchstander Requests

Approximately 150 bytes per second will result from messages from the watchstander to the main data base processor.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 60,000 bytes of data must be transmitted within a six second time period to allow two displays to be simultaneously replaced.

Status Messages

Status message will be sent periodically from processor to processor throughout the system. These messages will be used to determine the loading and current health of the processor. A rate of 500 bytes per second should be sufficient for this function.

Overhead

Message source and destination information, checksums and other overhead will be added to the interprocessor messages handled by the system. Positive acknowledgements may also be desirable. An overhead rate of 50% should be adequate.

10.4.3 Disc Sizing and Utilization

Moving head discs will serve as the mass storage devices for the data base discussed in Section 4. The discs will be identical to the discs used in the Class C, Level 4 system (see Section 10.3.3). Although actual disc storage requirements will be somewhat less than required for the Level C systems, the cost savings which might be achieved by using smaller discs is insufficient to justify using different discs.

Disc utilization for the Class B system will be identical to that calculated for the Class B system for the first architecture and shown in Figure 9-8. Total utilization is estimated to be .453 which indicates that a single disc will be sufficient to support the disc accessing. Two discs will be provided, however, to provide backup.

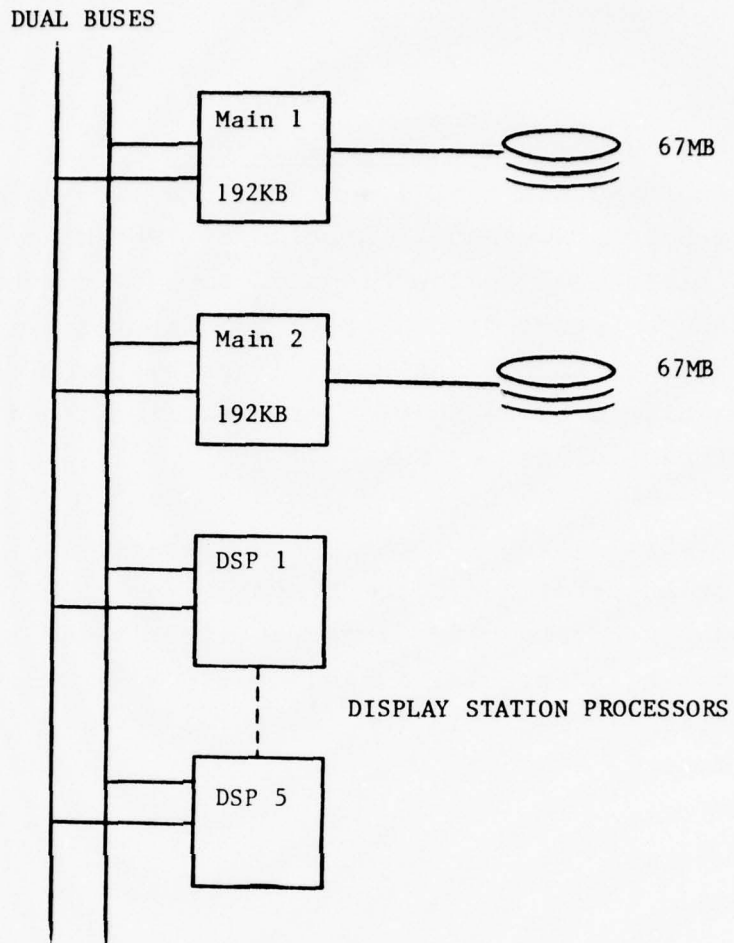


Figure 10-6. Shared Bus, Large Mini Architecture
100 Ships, Class A, Level 1

10.5 CLASS A, LEVEL 1 SYSTEM

In this section a preliminary design is presented for a Class A, Level 1 system capable of handling 100 vessels. The structure of this system is shown in Figure 10-6.

The system is almost identical to the one described for the Class B, Level 4 system (Section 10.4). The lesser number of functions allows the memory requirements to be reduced for the main processors and fewer display stations will be required.

10.5.1 Processor Sizing and Utilization

Five display stations will be required for the Class A, Level 1 system supporting 100 vessels. The display stations will be identical to those discussed previously. Two additional processors will be required. One will serve as the main processor performing all the on-line functions which are not handled by the display processors. The other will serve as an on-line backup.

Processing and memory requirements for the main processor are shown below. Processing loads for these processors assume no floating point hardware which further reduces the cost of the processor.

	<u>CYCLES/ SECOND</u>	<u>MEMORY K BYTES</u>
OS NUCLEUS	187,500	16
Bus Interface and Buffer		6
System Common		2
Data Base Buffer		8
File Handler	1,540	10
Overlay Handler		2

10.5 CLASS A, LEVEL 1 SYSTEM

In this section a preliminary design is presented for a Class A, Level 1 system capable of handling 100 vessels. The structure of this system is shown in Figure 10-6.

The system is almost identical to the one described for the Class B, Level 4 system (Section 10.4). The lesser number of functions allows the memory requirements to be reduced for the main processors and fewer display stations will be required.

10.5.1 Processor Sizing and Utilization

Five display stations will be required for the Class A, Level 1 system supporting 100 vessels. The display stations will be identical to those discussed previously. Two additional processors will be required. One will serve as the main processor performing all the on-line functions which are not handled by the display processors. The other will serve as an on-line backup.

Processing and memory requirements for the main processor are shown below. Processing loads for these processors assume no floating point hardware which further reduces the cost of the processor.

	<u>CYCLES/ SECOND</u>	<u>MEMORY K BYTES</u>
OS NUCLEUS	187,500	16
Bus Interface and Buffer		6
System Common		2
Data Base Buffer		8
File Handler	1,540	10
Overlay Handler		2

	<u>CYCLES/ SECOND</u>	<u>MEMORY K BYTES</u>
Overlay Area		12
Data Base Manager		10
Disc Driver		2
Memory Mapping		2
Route Stray	360	2
Dangerous Encounters	1,500	2
Excessive Congestion	6,000	2
Excessive Speed	900	2
Position Update	360	2
Encounters	9	2
Local Traffic	3,600	3
Environmental Information	85	2
Alert Responses	20,000	4
Simulation	79,620	4
Simulation Tables		20
System Backup Data Update	500	2
Vessel Passage History	10,000	2
VTs Operations Log	10,000	2
Notices Lookup	170	2
Search on a Key	100,000	6
Predefined Searches		2
Lane and Route Data		20
Passage Data		6
Alert Queue		4
Weather Data		1
History Buffers		12
Search Buffer		6
	<hr/>	<hr/>
	422,144	180

Processors with 192K bytes of memory will be required. CPU utilization is estimated at .388.

10.5.2 Interprocessor Communications Load

The interprocessor communications load will be slightly less than that calculated for the Class B, Level 4 system (section 10.4.2). The load generated when a display is replaced will predominate. A bandwidth of approximately 130,000 bits per second will be required.

10.5.3 Disc Utilization and Sizing

Discs will be of the same type as previously described for the Class A, Level 1 system for Candidate Architecture I (Section 9.5.3). The disc utilization of .212 as shown in Figure 9-11 will apply here as well.

10.6 RELATIVE HARDWARE COSTS

Relative hardware costs for Architecture II are summarized in Figure 10-6. As for Architecture I, these costs do not include the cost of hardware such as tape drives and printers which are common to all four architectures. The cost of interconnecting the display station processor is included since it is architecture dependent. The cost of the display station processor and display units is not included.

<u>Class C, Level 4</u>	<u>Number</u>	<u>Unit Cost</u> (<u>\$</u>)	<u>Total</u> (<u>\$</u>)
<u>Processors</u>			
128KB	1	27,000	27,000
192KB	2	34,000	68,000
320KB	2	48,000	96,000
<u>Discs</u>			
67MB	3	25,000	75,000
<u>Bus Couplers</u>	34	3,000	<u>102,000</u>
TOTAL			368,000
<u>Class B, Level 4</u>			
<u>Processors</u>			
320KB	2	48,000	96,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Bus Couplers</u>	18	3,000	<u>54,000</u>
TOTAL			200,000
<u>Class A, Level 1</u>			
<u>Processors</u>			
192KB*	2	29,000	58,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Bus Couplers</u>	14	3,000	<u>42,000</u>
TOTAL			150,000

*No floating point hardware

Figure 10-6. Cost Summary Shared Bus, Large Mini Architecture

The third candidate architecture is similar to the second architecture except in the method of interconnection.

As in the previous architecture (Section 10) the processors will be relatively high powered 16 bit minicomputers. For this architecture, dedicated serial lines will be used to interconnect the processors instead of a shared link.

11.1 CLASS C, LEVEL 4 SYSTEM

The structure of the Class C, Level 4 system is shown in Figure 11-1. Processors and discs are identical in function, size and utilization to those selected for Architecture II (See Section 10.3).

Processors will be interconnected by high speed synchronous serial lines. The bandwidth required for the interconnections will be discussed below.

Display Station Processors to Main Processors

The normal communications over this line consist of vessel position updates and the normal interactive messages resulting from watchstander inputs. While these communications loads can be substantial, they are small compared with the load generated when a complete new copy of the display is to be sent to the display station from the main processors. A rate of 40,000 bits per second was calculated for this load which implies the use of a line capable of approximately 50K bits per second.

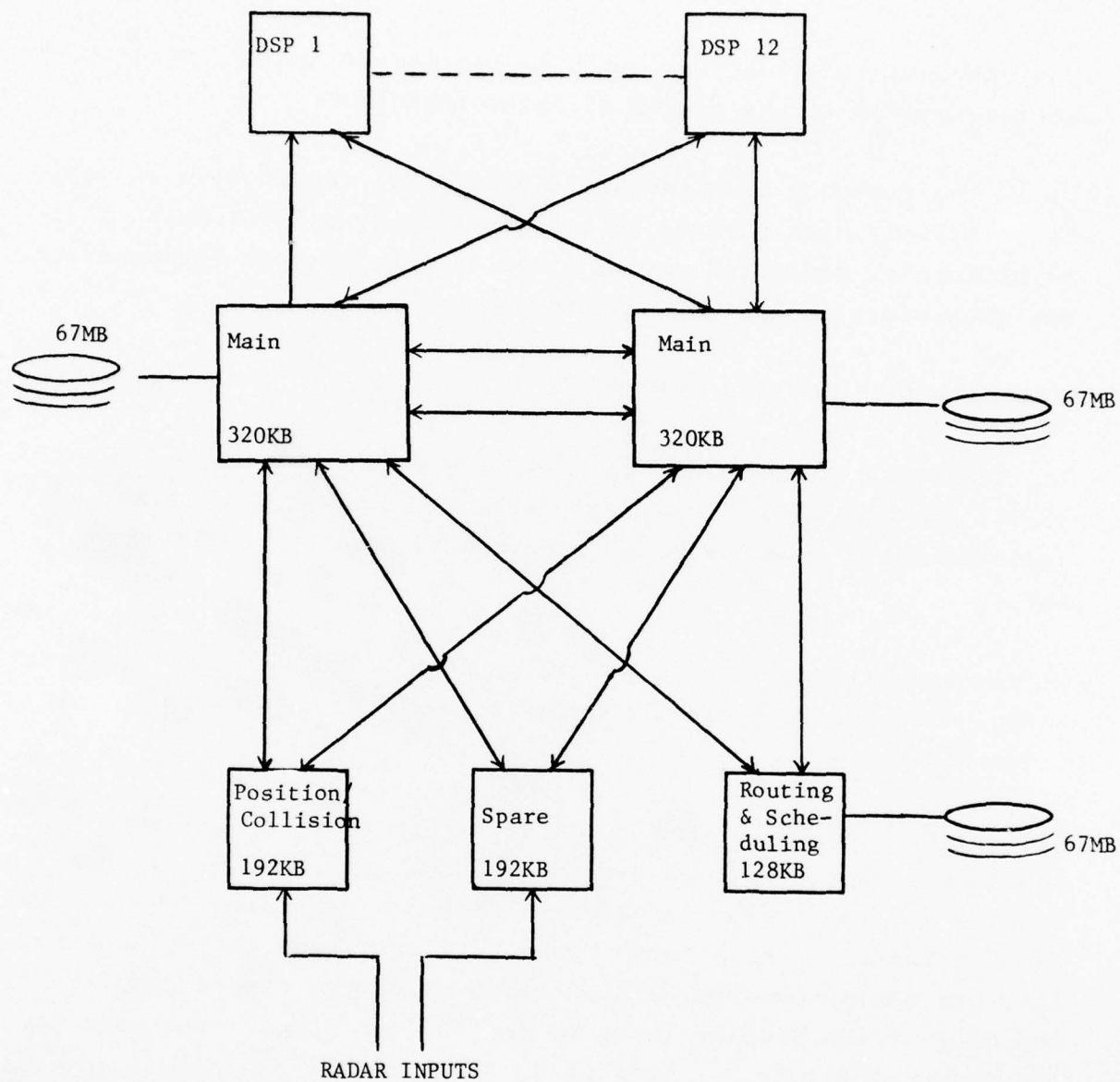


Figure 11-1. Dedicated Link, Large Mini Architecture
900 Ships, Class C, Level 4

Position/Collision Processor to Main Processors

The line joining the position/collision processor to the main processors must handle the transfer of a complete position table every 6 seconds. This load was estimated at approximately 175K bits per second in Section 10.3.2 implying the use of a line capable of approximately 200K bits per second.

Main to Main and Routing and Scheduling to Main

Communications between main processors and between the routing and scheduling processor and the main processor is low compared to the two cases just considered. The highest single load is the main to main data base transfer previously calculated to require approximately 24K bits/second.

11.2 CLASS B, LEVEL 4 SYSTEM

The structure of the Class B, Level 4 System is shown in Figure 11-2. Processor and disc sizing and utilization will again be identical to the equivalent system from Architecture II. (See Section 10.4)

Communications links will exist between the display stations and the main processors and between the two main processors. Approximate bandwidths required are 50K bits per second and 2K bits per second respectively.

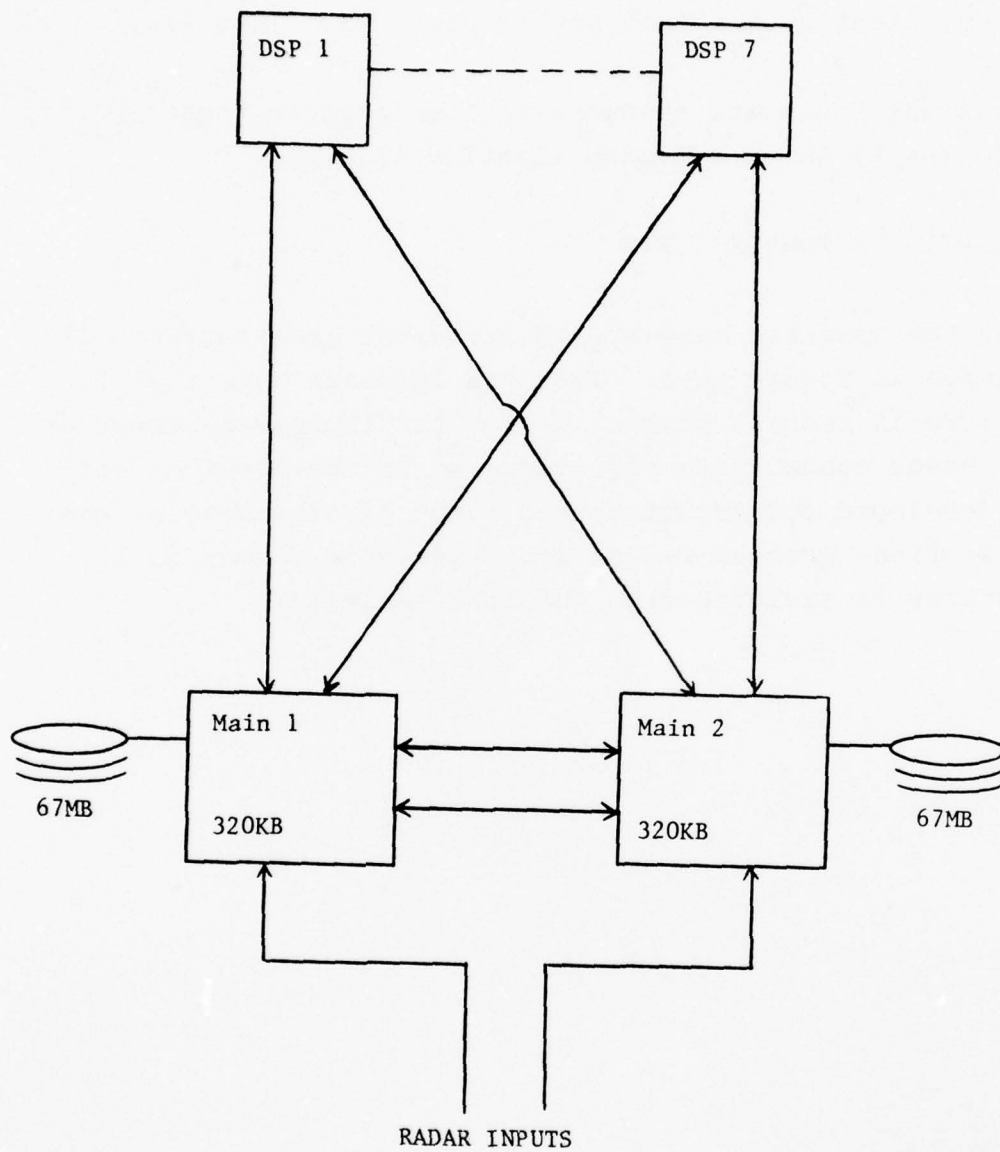


Figure 11-2. Dedicated Link, Large Mini Architecture
500 Ships, Class B, Level 4

11.3 CLASS A, LEVEL 1 SYSTEM

The structure of the Class A, Level 1 System is shown in Figure 11-3. Processor and disc sizing and utilizations will be identical to the equivalent system from Architecture II. (See Section 10.5)

Communications links are essentially identical to those specified for the Class B, Level 4 System (Section 11.2).

11.4 RELATIVE HARDWARE COSTS

Costs for the specific hardware required for Architecture III is presented in Figure 11-4. The cost increase over that for Architecture II results primarily from the increased number of interprocessor connections required. As in the previous cost figures developed for Architectures I and II, the cost of the display stations processors and other hardware common to all architectures is excluded from the cost estimate.

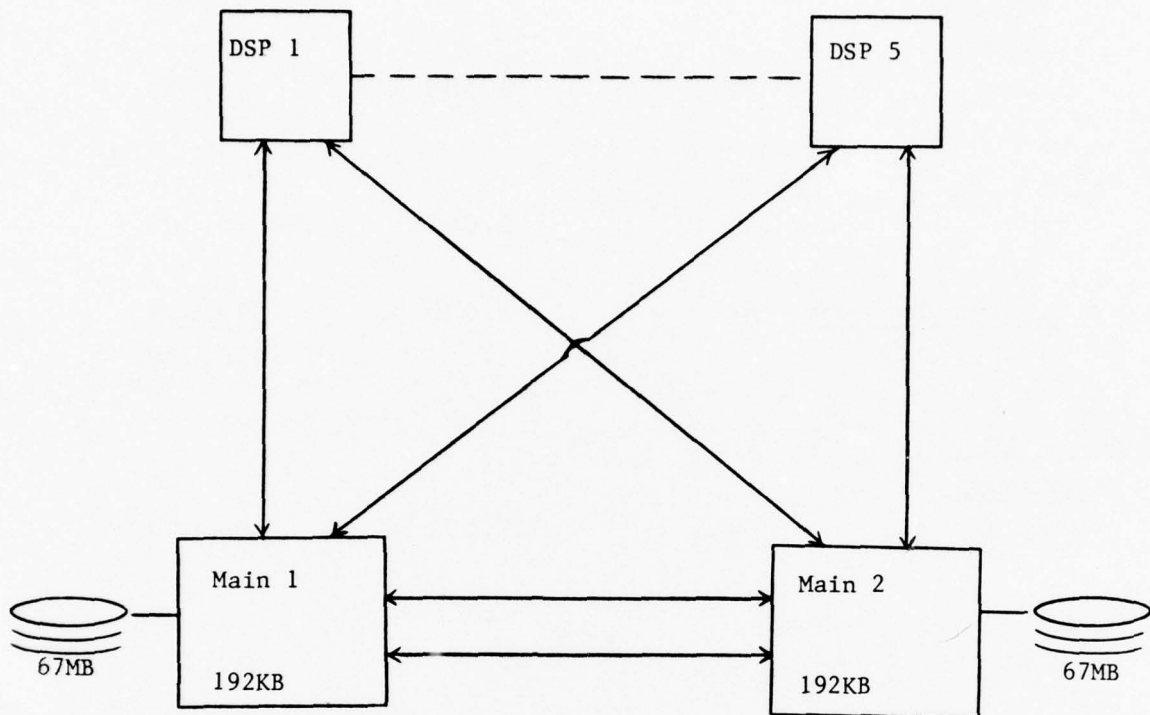


Figure 11-3. Dedicated Link, Large Mini Architecture
100 Ships, Class A, Level 1

	<u>Number</u>	<u>Unit Cost</u> (<u>\$</u>)	<u>Total</u> (<u>\$</u>)
<u>Class C, Level 4</u>			
<u>Processors</u>			
128KB	1	27,000	27,000
192KB	2	34,000	68,000
320KB	2	48,000	96,000
<u>Discs</u>			
67MB	3	25,000	75,000
<u>Sync. Line Drivers</u>	64	3,000	<u>192,000</u>
TOTAL			458,000
<u>Class B, Level 4</u>			
<u>Processors</u>			
320KB	2	48,000	96,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Sync. Line Drivers</u>	32	3,000	<u>96,000</u>
TOTAL			242,000
<u>Class A, Level 1</u>			
<u>Processors</u>			
192KB*	2	29,000	58,000
<u>Discs</u>			
67MB	2	25,000	50,000
<u>Sync. Line Drivers</u>	24	3,000	<u>72,000</u>
TOTAL			180,000

*No floating point hardware

Figure 11-4. Cost Summary Dedicated Link, Large Mini Architecture

The fourth candidate architecture uses a family of processors connected by a high speed shared link. Functional allocation is related to waterway sectors rather than grouping of software modules as in the previous architectures.

12.1 PHYSICAL STRUCTURE

The physical structure of the fourth candidate architecture is described below in terms of the processors and the interconnection mechanism.

12.1.1 The Processors

A family of both large and small processors is assumed for this architecture. The selection of a large processor (see Section 10.1.1 for characteristics) or a small processor (see Section 9.1.1 for characteristics) is based on the memory and processing requirements for the individual processor.

12.1.2 Interconnection

Processors will be interconnected by a shared link which will be duplicated to assure availability. The shared link could be an 8 or 16 bit bus or a shared serial line. The required bandwidth for the shared link will be discussed in Section 12.3.2.

12.2 LOGICAL STRUCTURE

Logically the system consists of a number of processors each performing a group of functions. Functional allocation for candidate architecture IV was designed to map physical processors to the functional elements which the VTS is required to support. The functions described in Appendix 8 - VTS Processing/Display Subsystem Function Description were divided into four types. This classification was based on which system entity the function supported. The four types correspond to the four basic system entities: The watchstanders, the vessel trackers (radars), logical VTS sectors and the overall VTS system itself.

The system is designed to simplify the task of configuring the hardware and software for new systems. Functional allocation is the same from system to system. To the maximum extent possible the hardware is identical from system to system although memory sizes are allowed to vary.

This approach allows new programs to be generated for the four functional types of processors based on the system parameters. Memory sizes can then be determined from the generated software. Hardware can then be assembled for the new system using the standard components.

12.3 CLASS C, LEVEL 4 SYSTEM

In this section a preliminary design is presented for a Class C, Level 4 system capable of handling 900 identified vessels. The structure of the system is shown in Figure 12-1.

12.3.1 Processor Sizing and Utilization

Functions have been assigned to processors in related functional groups. Four functional groups will be needed for the Class C, Level 4 system.

The following sections describe the processors which will be required for the 900 ship, Class C, Level 4 system. Memory sizes and CPU utilizations have been estimated for each of the processors and are summarized in Figure 12-2.

12.3.1.1 Position Input Processor

The position input processor will perform coordinate transformations and other radar support functions. All eight radars will be interfaced to the primary position input processor and to the backup processor.

Memory sizing and processor utilization for the position input processor is tabulated below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
Position Update	150,000	1
Tracker Tables 8 x 512 x 32 Bytes		128
OS Nucleus	125,000	16
Bus Interface and Buffers		6
Radar Interface Driver		2
Position Sensor Input Driver		1
System Common		2
Memory Mapping		2
TOTAL	<u>275,000</u>	<u>159</u>

AD-A061 307

INTERNATIONAL COMPUTING CO BETHESDA MD
PRELIMINARY DESIGN STUDY FOR VTS PROCESSING/DISPLAY SUBSYSTEM.(U)
JUN 78 C C HENSON, R A CLEAVER, S H KAISLER DOT-CG-81-78-1833
USCG-D-66-78 NL

UNCLASSIFIED

4 OF 4

AD
A061307



END
DATE
FILMED

2 - 79
DDC

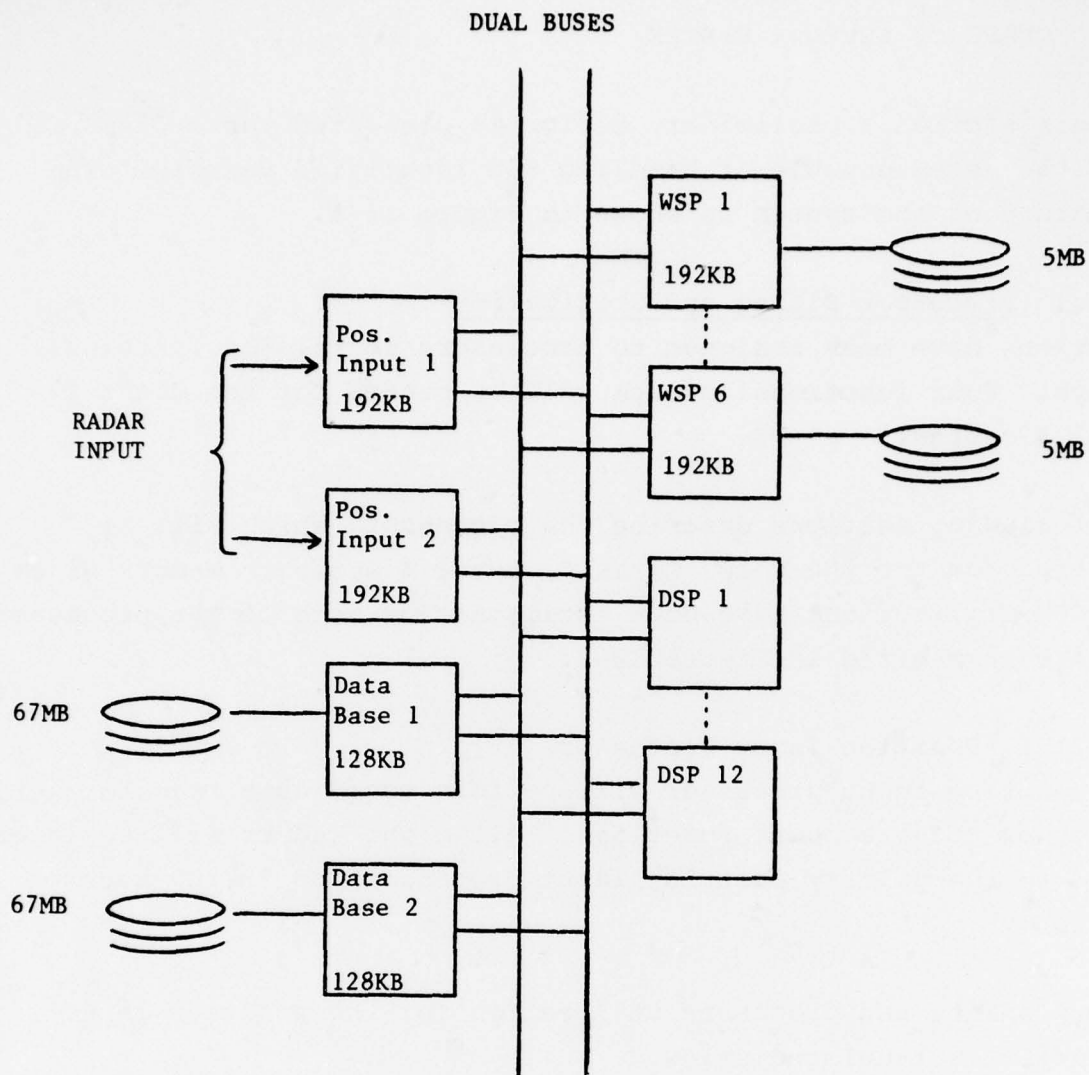


Figure 12-1. Shared Link, Sectorized Processing Architecture
900 Ships, Class C, Level 4

REFERENCE	PROCESSOR	NUMBER REQUIRED	---- MEMORY ----		CYCLES/ SECOND	CPU UTILIZATION
			REQUIRED K BYTES	ALLOCATED K BYTES		
12.3.1.1	Position Input	2	159	192	275,000	.220
12.3.1.2	Data Base	2	110	128	291,530	.486
12.3.1.3	Display Station	12				
12.3.1.4	Watchstander Support	6	151	192	747,404	.598

Figure 12-2. Processor Summary
900 Ship, Class C, Level 4

A large processor with 192K bytes of memory will be required to support the position input function. CPU utilization is estimated to be .220.

12.3.1.2 Data Base Processors

Two data base processors will be required to support data base and logging operations. Memory sizing and processing load is summarized below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
OS Nucleus	150,000	16
Bus Interface and Buffers		6
System Common		2
Data Base Buffers		20
File Handler	17,780	10
Overlay Handler		2
Overlay Area		12
Data Base Manager		10
Disc Driver		2
Memory Mapping		2
System Backup Data	3,750	2
Vessel Passage History	10,000	2
VTS Operations Log	10,000	2
Search On A Key	100,000	2
Predefined Searches		2
Search Buffer		6
History Buffers		12
	<hr/>	<hr/>
TOTAL	291,530	110

Small processors with 128k bytes of memory will be required for the data base processors. CPU utilization is estimated to be .486.

12.3.1.3 Display Station Processors

One processor will be required for each of the Display Stations. These processors will support the displays, manage editing for data entry functions and provide an interface between the display station and the remainder of the system. Section 5.3 discusses characteristics of these processors which are assumed to be the same for all the architectures studied.

12.3.1.4 Watchstander Support Processors

The system functions which are not handled by the position input, data base, and display station processors will be performed by the watchstander support processors.

All background hazard processing and all demand functions except those specifically assigned to the data base processors will be performed by the watchstander support processors.

Each watchstander support processor will handle the work associated with two sectors. Workload associated with a sector will vary from sector to sector. To determine the appropriate processing requirements we used the waterway model described in Section 3. Based on that model the worst case pair of sectors will contain about $1/3$ of the total ships or approximately 300 ships. For processing that is a direct function of the number of ships, $1/3$ of the total processing will be assumed.

Collision detection processing and the other processing which must consider all vessels in the immediate vicinity must deal with vessels in the two sectors plus vessels in the adjacent overlap areas. For these functions the watchstander support processor must be capable of handling approximately $1/2$ the total processing load for the overall system. Approximately 450 vessels would be considered.

Ten sectors will be required for the maximum configuration. Six watchstander support processors will be required with one of the six serving as a spare. The spare processor will support simulation when not required to replace a failed processor.

Memory sizing and processing load for the watchstander support processors is given below.

	<u>Cycles/ Second</u>	<u>Memory K Bytes</u>
OS Nucleus	187,500	16
Bus Interface and Buffers		6
System Common		2
File Handler	17,780	10
Overlay Handler		2
Overlay Area		6
Disc Driver		2
Memory Mapping		2
Lane Stray	10,000	2
Route Stray	10,000	2
Dangerous Encounters	4,500	2
Excessive Congestion	3,000	2
Anchor Drift	1,500	2
Navaid Adrift/Missing	3,300	2
Encounters	500	2
CPA	300	2
Local Traffic	10,800	3
Environmental Information	1,700	2
Alert Responses	6,700	4
Grounding and Speed Limit Checks	53,500	4
Notices Lookup	3,300	2
Grounding Buffer		6
Position Data 2,000 x 8 Bytes		16
Lane and Route Data		10
Basic Call Data		4
Passage Data 700 x 8 Bytes		6
Watch Circle Data		4
Alert Queue		2
Weather Data		1
Collision - Stage 1	242,730	3
Stage 2	900	
Stage 3	1,894	
Routing and Scheduling	187,500	6
Buffer for Scheduling		16
TOTAL	<u>747,404</u>	<u>151</u>

Large processors with 192K bytes of memory will be required. CPU utilization is estimated at .598.

12.3.2 Interprocessor Communication Loads

An estimate of the interprocessor communications load has been made to provide an approximation of the bandwidth required for the bus or other shared link. These loads are summarized in Figure 12-3 and discussed below.

	<u>BYTES</u>	<u>TIME SECONDS</u>	<u>BYTES PER SECOND</u>	<u>BITS PER SECOND</u>
Position Input Data	80,000	6	13,333	106,664
Data Base Writes	1,464	1	1,464	11,712
Watchstander Requests	300	1	300	2,400
Display Replace	60,000	6	10,000	80,000
Routing and Scheduling	84,450	1	84,450	675,600
				<hr/> 876,376
Overhead - 50%				438,188
				<hr/> <hr/>
TOTAL				1,314,564

Figure 12-3. Interprocessor Communications Load

Position Input Data

Position input data for all vessels and buoys which are in track will be transmitted from the position/input processor to the watchstander support processors. In determining the communications load we have assumed that position data for 1/2 the total vessels and buoys will be transmitted to each watchstander support processor every 6 seconds. Eight bytes will be transmitted for each of 2,000 entries. Five copies will be transmitted for a total load of 80,000 bytes every 6 seconds.

Data Base Writes

Copies of the data to be written to the disc must be sent from the main data base processor to the secondary data base processors. Assuming 1.43 writes per second (Figure 4-11) at 1024 bytes per write this gives a load of 1464 bytes per second.

Watchstander Requests

Approximately 300 bytes per second can be assumed as the peak for messages from the watchstanders to the main data base processor.

Display Replace

Periodically, displays will be replaced to show a different magnification or to change to a display of a different area of the harbor. Approximately 30,000 bytes of data must be transmitted within a six second time period for each display replaced.

The computed total load assumes that two display replacements can occur at any given time. This assumption is reasonable since the frequency of display replacements is expected to be low. It should be noted, however, that the bandwidth required of the processor interconnection would be increased significantly if provision is made to allow more than two displays to be replaced simultaneously.

Status Messages

Status messages will be sent periodically from processor to processor throughout the system. These messages will be used to determine the loading and current health of the processors. A rate of 1,000 bytes per second should be sufficient for this function.

Routing and Scheduling Data

Routing and scheduling data will be transmitted as needed from the data base processor to the watchstander support processors. Approximately 84,450 bytes per second are required.

Overhead

Message source and destination information, checksums and other overhead will be added to the interprocessor messages handled by the system. It may also be desirable to send positive acknowledgements to each message. An overhead rate of 50% should be adequate.

12.3.3 Disc Sizing and Utilization

Two types of disc will be used with this architecture. The main discs will be identical to those selected for candidate architecture I (see Section 9.3.3).

The watchstander support processors will each have a 5 megabyte disc which will support grounding and will allow overlays to be loaded for miscellaneous functions which are not resident in main memory.

Typical characteristics of these discs are:

- . 5 megabytes of storage
- . 25 ms revolution time
- . 40 ms to locate a cylinder
- . 6,144 bytes per track

If the grounding scan is performed by reading one track at a time, utilization of these discs will be approximately .538.

Utilization factors for the other functions, all of which will be serviced by the main discs, will be the same as those calculated in Section 9.3.3.1 and summarized in Figure 9-4. When both main discs are functioning, one disc would support data base searches and all normal reads and writes. Total utilization would be .478.

The second main disc would support routing and scheduling and maintain a current copy of the entire data base. Utilization would be .340.

12.4 CLASS B, LEVEL 4 SYSTEM

The Class B, Level 4 system capable of supporting 150 identified and 350 unidentified vessels can be easily generated using the same basis as the Class C, Level 4 system.

The functional processors are the same. Memory requirements are reduced, however. The structure of the system is shown in Figure 12-4.

Position input processors will be functionally identical to those provided for the Class C, Level 4 system. Since three radars instead of eight will be supported, tracker table space is reduced so that actual memory requirements will be reduced to 79K bytes. Processors with 128K bytes of memory will be assumed.

Display station processors and data base processors will be identical to those supplied for the Class C system.

The watchstander support processors for all levels of system will each support two sectors. This fact simplifies the task of configuring a new system. Because routing and scheduling are not required in a Class B system and table space can be reduced somewhat, the watchstander support processors will require only 128K bytes of memory.

The interprocessor links and system discs are assumed to be identical to those for the larger Class C system. The number of 5 megabyte discs will be reduced however since only 4 watchstander support processors will be required.

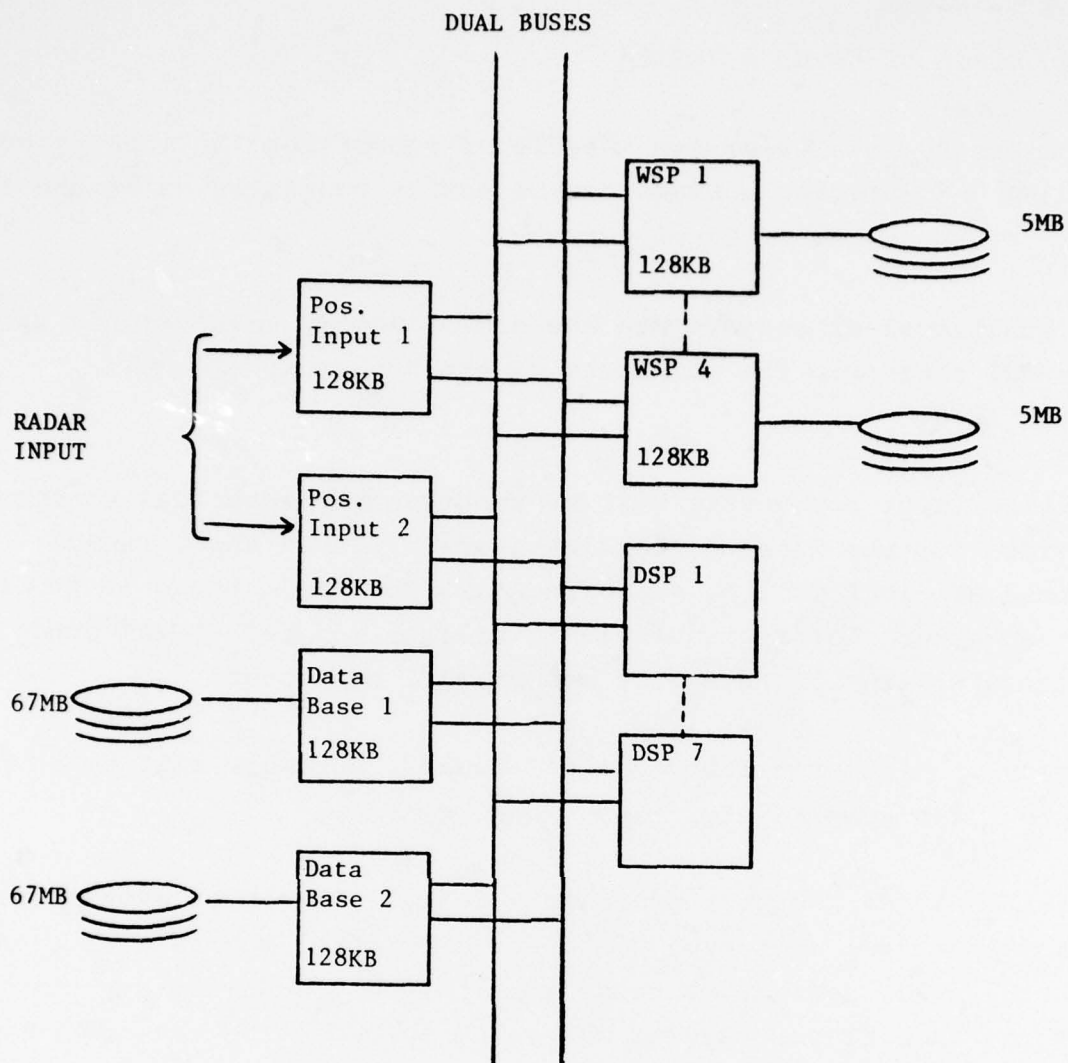


Figure 12-4. Shared Link, Sectorized Processing Architecture
500 Ships, Class B, Level 4

12.5 CLASS A, LEVEL 1 SYSTEM

The Class A, Level 1 system supporting 100 vessels is configured in the same manner as the Class B system. Figure 12-5 shows the configuration.

Position input processors are eliminated since radar is not included. Hardware for the watchstander support processors and data base processors will be identical to the Class B system except that less watchstander support processors will be required.

The interprocessor links and system discs will be identical to those specified for the other systems.

12.6 RELATIVE HARDWARE COSTS

Relative hardware costs for Architecture IV are summarized in Figure 12-6. As for the other architectures, these costs do not include the cost of hardware such as tape drives, printers and display station processors which are common to all four architectures.

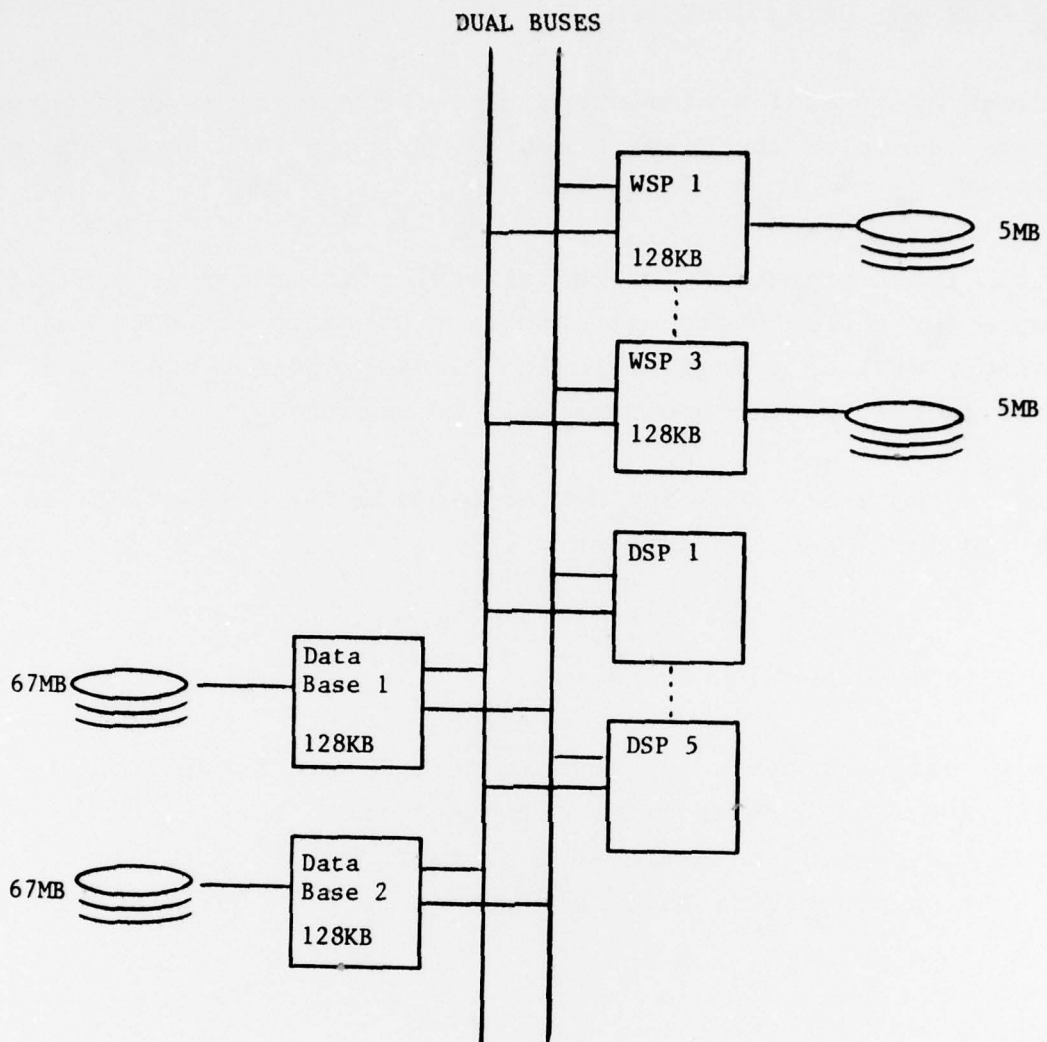


Figure 12-5. Shared Link, Sectorized Processing Architecture
100 Ships, Class A, Level 1

	<u>NUMBER</u>	<u>UNIT COST</u>	<u>TOTAL</u>
<u>CLASS C, LEVEL 4</u>			
<u>Processors</u>			
192KB	8	34,000	272,000
128KB*	2	18,000	36,000
<u>Discs</u>			
67MB	2	25,000	50,000
5MB	6	11,000	66,000
<u>Bus Couplers</u>	44	3,000	132,000
			<hr/> 556,000

<u>CLASS B, LEVEL 4</u>			
<u>Processors</u>			
128KB	6	27,000	162,000
128KB*	2	18,000	36,000
<u>Discs</u>			
67MB	2	25,000	50,000
5MB	4	11,000	44,000
<u>Bus Couplers</u>	30	3,000	90,000
			<hr/> 382,000

<u>CLASS A, LEVEL 1</u>			
<u>Processors</u>			
128KB**	3	22,000	66,000
128KB*	2	18,000	35,000
<u>Discs</u>			
67MB	2	25,000	50,000
5MB	3	11,000	33,000
<u>Bus Couplers</u>	20	3,000	60,000
			<hr/> 145,000

*600,000 CPS processor
 **No floating point hardware

Figure 12-6. Cost Summary
 Shared Link, Sectorized Processor Architecture

The four architectures presented in Sections 9 through 12 have all been shown to be viable alternatives for the VTS Processing/Display Subsystem. Evaluating the alternatives is not straight forward since no one architecture is clearly the "best" based on the criteria presented in Section 8. Each architecture has its own unique strengths and weaknesses.

The evaluation is further complicated by the fact that many of the criteria which are appropriate for evaluating architectures are subjective and therefore subject to individual biases.

In this chapter we will present the methodology used to consider and correlate the various factors to arrive at a relative ranking of the four architectures. Based on this evaluation, Architectures II and IV have been judged the best with Architectures III and I following close behind.

13.1 EVALUATION METHODOLOGY

The approach used to evaluate the candidate architectures and select two architectures for the VTS system consisted of three steps. In step 1, the project team determined the relative weighting factors for the eight selection criteria described in Section 8. The weight for each selection criteria was further subdivided into weights for hardware and software factors. The sum of the weights for the eight selection criteria totals 100 points. The sum of the hardware and software weights for each criterion also totals 100 points. The individual weights for each criterion and for the hardware and software factors are displayed in Figures 13-1.

In step 2, each member of the project team evaluated all four candidate architectures using the selection criteria described in Section 8. An individual score was given for both hardware and software factors for each criterion. The individual scores were then averaged for each factor and transferred to the evaluation sheet. This table is displayed in Figure 13-2.

In step 3, the hardware and software factors for each criterion were added together and transferred to the summary sheet. The raw sum for each architecture was computed. The figure of merit (FOM) was computed by the formula

$$FOM_j = (\sum_{i=1}^8 c_{ij} \omega_i) / 100, \quad j = 1, 4$$

where c_{ij} = value of i th criterion for j th architecture
 ω_i = value of i th weight
 FOM_j = value of FOM for j th architecture

<u>CRITERION</u>	<u>HARDWARE</u>	<u>SOFTWARE</u>	<u>WEIGHT</u>
Simplicity	40	60	15
Feasibility	75	25	15
Modularity	35	65	15
Maintainability	30	70	5
Expandability	40	60	10
Reliability	40	60	10
Cost	25	75	20
Performance	60	40	<u>10</u>
			100

Figure 13-1. Weighting Factors for Evaluation Criteria

ARCHITECTURE EVALUATION SHEET

CRITERION	ARCHITECTURE SCORE			
	I	II	III	IV
I. Simplicity				
Hardware	23	37	33	28
Software	38	57	55	48
II. Feasibility				
Hardware	40	53	75	45
Software	12	20	25	20
III. Modularity/Flexibility				
Hardware	35	23	18	30
Software	52	53	53	58
IV. Maintainability				
Hardware	22	30	23	22
Software	70	52	47	63
V. Expandability				
Hardware	40	30	22	35
Software	38	55	47	55
VI. Reliability				
Hardware	38	33	25	38
Software	55	50	35	55
VII. Cost				
Hardware	20	25	15	10
Software	55	72	58	70
VIII. Performance				
Hardware	58	58	58	58
Software	38	32	27	40

Figure 13-2. Architecture Evaluation Sheet

The totals for each architecture evaluation criterion and the figure-of-merit for each architecture are shown in Figure 13-3. Based on the figures-of-merit, Architectures II and IV, in that order, appear to be the most effective with Architectures III and I following, in that order.

CRITERION	ARCHITECTURE TOTAL SCORES			
	I	II	III	IV
I. Simplicity	61	94	88	76
II. Feasibility	52	73	100	65
III. Modularity/Flexibility	87	76	71	88
IV. Maintainability	92	82	70	85
V. Expandability	78	85	69	90
VI. Reliability	93	83	60	93
VII. Cost	75	97	73	80
VIII. Performance	96	90	85	98
Raw Sum	634	680	616	675
Figure-of-Merit	76	86	78	83

Figure 13-3. Architecture Evaluation Summary

13.2 DISCUSSION OF WEIGHTING FACTORS

The project team assigned relative weights to hardware/software factors and individual criterion based on our experience in the design and development of distributed architectures. The evaluation of the candidate architectures required both subjective and objective analysis. The objective analysis of each architecture has been discussed in Sections 9 through 12. This analysis was based on data supplied by the U. S. Coast Guard as it applied to the requirements discussed in the VTS Processing/Display Subsystem Functional Description. The subjective analysis was based on previous experience with similar systems.

In order to complement the evaluation methodology described above, a brief discussion of the rationale for assigning the weighting factors for each criterion is presented below.

Simplicity

We assigned a relative weight of 15 to the criterion for simplicity. Because many VTS systems may be built it was felt that a simple architecture would be more readily adaptable to the different waterways and different class/level configurations in which a VTS could operate. The hardware/software factors were weighted at 40/60 respectively. ICC felt that software deserved a higher rating because of the additional effort required to generate a VTS system from the baseline modules.

Feasibility

We assigned a relative weight of 15 to the criterion for feasibility. The architecture should be capable of implementation from off-the-shelf components or with minimal hardware development. The hardware/software factors were rated at 75/25 respectively. Hardware is given

a 3 to 1 weight over software since it is more difficult to adjust a hardware configuration once the system has been selected.

Modularity

We assigned a relative weight of 15 to the criterion for modularity and flexibility. It is important that any architecture selected be adaptable to a wide range of waterways. The hardware architecture must also provide easy transition to different class/level configurations as the traffic characteristics of the waterway change. The hardware/software factors were rated at 35/65 respectively.

Maintainability

We assigned a relative weight of 5 to the criterion for maintainability. It is difficult to evaluate architectures for this criterion because many of the factors are related to policies and procedures. Software maintainability is usually more difficult and contributes extensively to life cycle costs. Thus the hardware/software factors were rated at 30/70 respectively.

Expandability

We accorded this criterion a relative weight of 10 because many of the factors overlap with the criterion for modularity. However, this criterion was an estimate of growth within a given class/level configuration and for a given VTS installation. Since software changes are likely after a system has been installed, the hardware/software factors were rated at 40/60 respectively.

Reliability

We accorded this criterion a relative weight of 10. While reliability is an important criteria, all architectures considered are designed to provide hardware reliability. The estimate of software reliability

is difficult to make without a detailed software architecture. The hardware/software factors were rated at 40/60 because of the greater complexity and thus greater potential for faults in the software.

Cost

We assigned a relative weight of 20 to the criterion for cost. We believe this to be the most important criterion in evaluating architectures since it implicitly includes all other criteria. Many of the factors associated with cost cannot be determined until a more detailed configuration and software specifications are developed. We made rough subjective estimates of costs over the life cycle of the system based on the assumption that software/personnel costs will run in a ratio of 3 to 1 over hardware costs.

A summary of the hardware costs which are unique to the individual architectures is included in Figure 13-4. The costs shown do not include the cost of display stations which are assumed to be the same for all architectures. It does include the cost of connecting the display station processors to the balance of the system.

Magnetic tapes, printers, and other peripherals are not included and are assumed to be the same for all architectures.

Performance

We accorded this criterion a weight of 10. Based on performance analyses for critical processing tasks (see Section 3), ICC determined a value for each architecture. The candidate architectures were initially culled from the set of feasible architectures based on a subjective estimate of performance. This estimate and analysis were further refined for each architecture for both hardware and software.

	ARCHITECTURE			
	I	II	III	IV
<u>CLASS C, LEVEL 4</u>				
Processors	\$247,000	\$191,000	\$191,000	\$308,000
Discs	75,000	75,000	75,000	116,000
Interconnection	<u>180,000</u>	<u>102,000</u>	<u>192,000</u>	<u>132,000</u>
	\$502,000	\$368,000	\$458,000	\$556,000
<u>CLASS B, LEVEL 4</u>				
Processors	\$105,000	\$ 96,000	\$ 96,000	\$198,000
Discs	50,000	50,000	50,000	94,000
Interconnection	<u>84,000</u>	<u>54,000</u>	<u>96,000</u>	<u>90,000</u>
	\$239,000	\$200,000	\$242,000	\$382,000
<u>CLASS A, LEVEL 1</u>				
Processors	\$ 54,000	\$ 58,000	\$ 58,000	\$102,000
Discs	50,000	50,000	50,000	83,000
Interconnection	<u>48,000</u>	<u>42,000</u>	<u>72,000</u>	<u>60,000</u>
	\$152,000	\$150,000	\$180,000	\$245,000

Figure 13-4. Hardware Cost Comparisons

APPENDIX

Hardware Cost Factors

The purpose of this Appendix is to specify the hardware cost factors used in Sections 9 through 12 to estimate hardware costs on a common basis. The hardware cost estimates were developed for relative comparison of architectures. Hardware common to all architectures, such as tape drive(s) and printer(s) was not included in the cost estimate. Thus, the cost figures are not representative of the total cost of a particular VTS system.

The hardware items of major significance in VTS architecture cost comparisons are processors (and associated memory) and disc drives. To provide realistic cost comparisons, we examined prices for processors and disc drives of various sizes and speeds from a number of different manufacturers. Two processor speeds were considered with cycle times of 1.6 sec and 800nsec. Disc drives with storage capacities between 5mb and 128mb were considered. In tabulating processor costs, we included, where applicable, features such as power fail detect/automatic restart, automatic program load, memory mapping, parity memory, and expansion chassis. For the 800nsec processor \$5000 is the approximate cost of floating point hardware.

Figure A-1 presents the results of the hardware cost analysis. The prices shown represent the mean of the prices considered, smoothed to an integral multiple of \$1000. The price of the processors is the list price, and varies with the amount of memory assumed.

	<u>Approximate Cost</u>
<u>1.6μsec Processors</u>	
64KB	\$11,000
128KB	18,000
<u>800nsec Processors</u>	
64KB	15,000
128KB	22,000
192KB	29,000
256KB	36,000
320KB	43,000
384KB	50,000
For Floating Point (800ns processors only) Add: \$5,000	
<u>Discs</u>	
5MB	11,000
10MB	13,000
25MB	24,000
67MB	25,000
128MB	35,000

Figure A-1. Hardware Cost Factors

*U.S. GOVERNMENT PRINTING OFFICE: 1978-700-504/207